sintax

para usuarios TIMEX SINCLAIR - TK 83 y 85



Nº 3

oct-1984

Indice

ECUACIONES DIFERENCIALES

```
  10 REM   EC-DIFF
      ECUACIONES DIFERENCIALES
      MAGNASCO, M.O. 1984

  20 PAPER 0: INK 7: BORDER 1: P
RINT '''" Resolucion numerica de
 una ecuacion diferencial de 2o
orden"''" x""+a(x',t)+b(x,t)=h(t
)"''
 100 INPUT " Entre x""en funcion
 de x, x' y t x""(X,X1,T)=";F$
 110 INPUT " Entre t en funcion
de x""        t(X2)=";T$
 120 INPUT "Condiciones iniciale
s            x0=";X,"x'0=";X1
 130 CLS : PRINT " Desea el graf
ico de x en fun-  cion del tiemp
o, en el espacio  de las fases,
o solo la impre-  sion de los va
lores?"''" Marque G para grafico
, F para fases o I para impresio
n"'',"_"
 140 LET i$=INKEY$: IF i$="" THE
N GO TO 140
 150 IF i$<>"F" AND i$<>"G" AND
i$<>"I" THEN PRINT AT 8,16;i$;"?
_": GO TO 140
 160 LET GRA=0: LET IMP=0: LET F
ASE=0: IF i$="G" THEN PRINT AT 8
,16;"GRAFICA": LET TT=0: LET GRA
=1: INPUT "ET";ET,"EX";EX: CLS :
 DRAW 0,175: PLOT 0,88: DRAW 255
,0
 170 IF I$="F" THEN PRINT AT 8,1
6;"FASES": INPUT "EX";EX,"EX'";E
V: CLS : LET FASE=1: PLOT 0,88:
DRAW 255,0: PLOT 128,0: DRAW 0,1
75
 180 LET T=0: IF I$="I" THEN LET
 IMP=1: PRINT AT 8,16;"IMPRIMIR"
  BEEP .1,0: CLS
 190 LET X2A=VAL F$: REM

 200 LET DT=VAL T$: LET X2=VAL F
$: LET X=X+X1*DT+(4*X2-X2A)*DT*D
T/6: LET X1=X1+(3*X2-X2A)/2*DT:
LET T=T+DT: LET X2A=X2
 210 REM

 220 IF FASE THEN PLOT X*EX+128,
X1*EV+88
 230 IF IMP THEN PRINT T;TAB 5;X
;TAB 20;X1
 240 IF GRA THEN LET TT=TT+ET*DT
  PLOT TT,X*EX+88: IF TT>=253 TH
EN CLS : DRAW 0,175: DRAW 0,-88:
 DRAW 255,0: LET TT=0
 250 GO TO 200
```

Es un programa para aproximar la solucion de una ecuacion diferencial de segundo orden, $F(x,x',x'',t)=0$, donde x'' puede ponerse en funcion del resto de los parametros, $x''=F(x,x',t)$ y doy x_0 , x'_0 . El metodo es muy simple: desarrollo x en una serie de Taylor alrededor de O:

$x(\Delta t)=x_0 + x'_0 * \Delta t)+(x''_0 * \Delta t**2)/2+(x'''_0 * \Delta t**3)/6$ $x''_0 = F(x_0, x'_0, 0)$

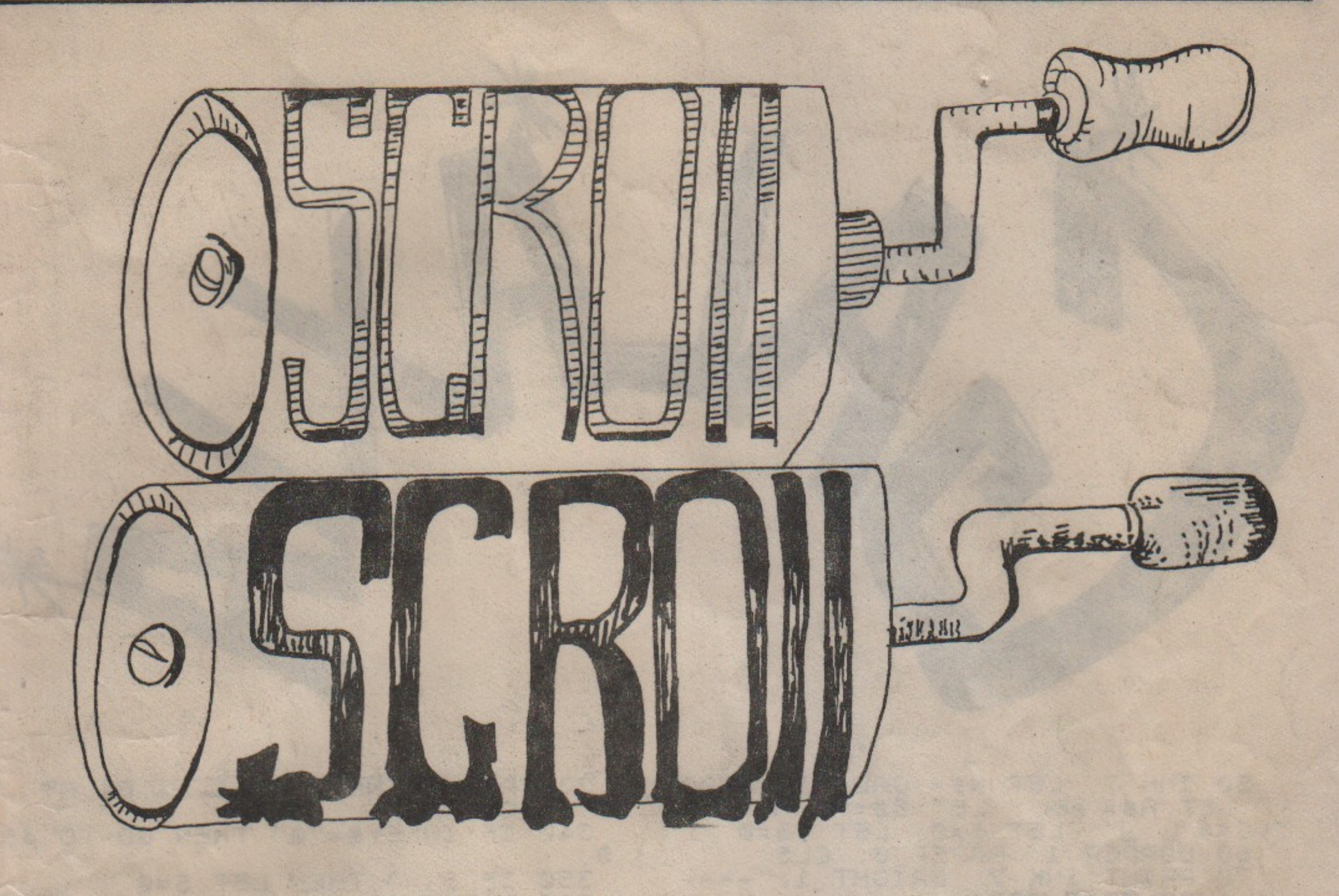
$x'(\Delta t)=x'_0 + x''_0 (\Delta t)+x'''_0 (\Delta t**2)/2$ $x'''_0 = -(x''_0 - x''(\Delta t)/\Delta t$

$x(\Delta t)=x_0 + x'_0 * \Delta t+(4*x''_0 - x'')*\Delta t**2/6$

Obtenido $x(\Delta t)$, $x'(\Delta t)$, hago un desarrollo identico tomando como origen Δt, y asi sucesivamente. Esta aproximacion es correcta hasta el segundo orden en Δt.

Ejemplo:

x2="-x-0.1*x1"
Δt="0.02"
$x_0=60$, $x'_0=0$
rango de x=1
rango de x'=1



LISTADO 1

```
   1 REM 111111111122222222222333
3333333444444444445555555555566666
6666677777777777888888888889999999
9990000000000011111111111222222222
2333333
   2 LET S1=16626
   3 LET S2=16514
   4 LET S3=16633
   5 LET S4=16560
   6 LET S6=16588
   7 LET S7=16619
   8 LET S8=16629
   9 LET S9=16612
  10 LET A$="2A0C40E511210019D10
1D602ED60C92A1040114300ED52E5112
100ED52D101B502EDB82A0C400620233
50010FBC92A0C4011D6021906162B4E3
E002B7EFE762802180310F2C9714F18F
12A0C400616234E3600237EFE7628021
80310F2C9714F18F1CDCC40CD9140C9C
DB040CD9140C9CDB040CD8240C9CDCC4
0CD8240C9"
  19 FAST
  20 LET I=16514
  30 FOR J=1 TO LEN A$ STEP 2
  40 POKE I,(CODE A$(J)-28)*16+C
ODE A$(J+1)-28
  50 LET I=I+1
  55 IF PEEK I=118 THEN GOTO 70
  60 NEXT J
  70 SLOW
```

LISTADO 2

```
  10 LET A$="50"
  20 LET A$(2)=STR$ (VAL A$(2)+.

  30 IF A$(2)="5" THEN GOTO 20
  40 FOR I=1 TO 22
  50 PRINT "===USUARIO Y PROG. D
E SINCLAIR==="
  60 NEXT I
  70 FOR I=1 TO 32
  80 RAND USR VAL A$
  90 NEXT I
 100 CLS
 110 GOTO 20
```

Es un programa con codigo de maquina para mover el display cruzando la pantalla en cualquiera de las 8 direcciones cardinales.

Entre el listado 1, tecleando los 126 caracteres en la sentencia REM de la linea 1. Corra el programa y el codigo de maquina sera pokeado por la sentencia REM .

Deletee luego de la linea 10 a la 70 y reemplacelas por la rurina de demostracion del listado 2.

Si entra el comando "GO TO 10" la versatilidad del programa SCROLL sera demostrada.




```
  50 INK 7: LET N$="GALAXIA 2000
": LET A$="⊧⊜⊨": LET B$="⌜ ⌝": LE
T C$="⌞ ⌟": LET S=0: LET TS=0
  60 BORDER 1: PAPER 0: CLS
  70 PRINT INK 7; BRIGHT 1;"----
-----GALAXIA 2000-----------"
. 80 PRINT AT 13,0;"EL MAXIMO PU
NTAJE LO POSEE ": PRINT : PRINT
 n$;" EN ";TS;" PUNTOS"
  90 PRINT AT 20,0; FLASH 1;"Pul
se cualquier tecla para comenzar
"
 101 PRINT AT 3,0;"Pulsando las
flechas del cursor    mueve su n
ave,cuando tiene      centrada en
 la mira la nave      invasora pu
lse o (CERO)para      disparar el
 laser"
 140 PAUSE 0
 150 REM INICIO
 160 CLS
 170 PRINT AT 8,14;B$
 180 PRINT AT 13,14;C$
 190 LET Q=INT (RND*16)+2: LET R
=INT (RND*26)+2
 200 PLOT 0,15: DRAW 255,0: FOR
X=0 TO 40 STEP 4: PLOT 125-X,15:
 DRAW -2*X,-15: PLOT 125+X,15: D
RAW 2*X,-15: NEXT X: LET Q=13: F
OR X=73 TO 125 STEP 5: PLOT X,15
: DRAW -X,-Q: LET Q=INT (Q/6)*5:
 NEXT X: LET Q=13: FOR X=178 TO
255 STEP 5: PLOT X,15: DRAW 255-
X,-Q: LET Q=INT (Q/6)*5: NEXT X
 220 FOR t=500 TO 0 STEP -1
 230 PRINT INK 7; BRIGHT 1;AT Q,
R;A$
 240 LET V=Q: LET W=R
 250 LET Q=Q+(INT (RND*2)-1): LE
T R=R+(INT (RND*2)-1)
 260 LET R=R+(2*(INKEY$="5")-(IN
KEY$="8"))
 270 LET Q=Q+(2*(INKEY$="7")-(IN
KEY$="6"))
 280 IF Q<=3 THEN LET Q=Q+3
 290 IF R<=3 THEN LET R=R+3
 300 IF Q>=18 THEN LET Q=Q-3
 310 IF R>=28 THEN LET R=R-3
 320 LET I=6: IF Q>9 AND Q<12 AN
D R>12 AND R<16 THEN LET I=2
 330 PRINT INK 7;AT 8,14;B$,AT 1
3,14;C$
 340 IF INKEY$="0" THEN GO TO 48
0
 350 IF S<=0 THEN LET S=0
 360 PRINT AT V,W;"  "; INK 1; P
APER 6;AT 0,0;" Puntuacion=";S;"
 "; INK 0; PAPER 5;"  Tiempo=";T
;" "
 370 BEEP .0025,5: BEEP .0025,6
 380 NEXT T
 390 GO SUB 650
 400 PRINT INK 6; FLASH 1;AT 10
10;"FIN"; INK 2;" Puntos=";S
 410 IF S>TS THEN GO SUB 600
```



```
420>BORDER 1: PRINT AT 20,2; IN
 3;"Presione cualquier tecla pa
a jugar de nuevo."
430 LET S=0
440 FOR X=0 TO 500
450 IF INKEY$<>"" THEN GO TO 60
460 NEXT X
470 GO TO 50
480 BEEP .01,7: FOR P=2 TO 6 ST
EP 2: INK P: PLOT 10,16: DRAW 11
2,72: DRAW 120,-72: PLOT 11,16
DRAW 112,71: DRAW 120,-71: NEXT

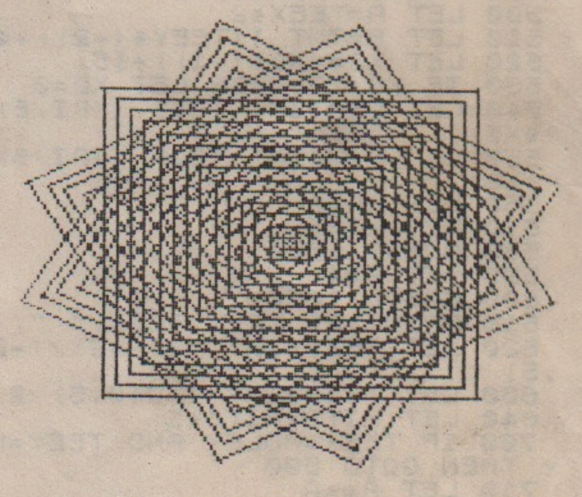
490 BEEP .015,9: BEEP .015,8
500 IF Q>9 AND Q<12 AND R>12 AN
D R<16 THEN GO TO 530
510 LET S=S-20
520 GO TO 580
530 PRINT PAPER 6; INK 2; FLASH
1;AT Q,R-1;"▒▒▒▒";AT Q+1,R;"▒▒
AT Q-1,R;"▒▒"
540 FOR X=0 TO 10: BEEP .005,4
BEEP .0,9: NEXT X
550 BEEP .01,7: BEEP .015,2
560 FOR X=0 TO 50: NEXT X: PRIN
T AT Q,R-1;"      ";AT Q-1,R;"  "
AT Q+1,R;"  ": LET Q=INT (RND*15
)+2: LET R=INT (RND*26)+2
570 LET S=S+50
580 FOR P=19 TO 11 STEP -1: PRI
NT AT P,0;"
          ": NEXT P
590 GO TO 350
600 FOR X=0 TO 150:::::: BORDE
R 2: BORDER 6: NEXT X: INPUT ; F
LASH 1;"MAXIMA PUNTUACION "; FLA
SH 0;"Escribe tu nombre (max.9 l
etras)"; LINE n$
610 IF LEN n$>=10 THEN GO TO 60
0
```

NOTA GRAFICA:

Para dibujar la nave: GRAPHICS "AB". Los graficos de B$ son: GRAPHICS "D E", y los de C$ son GRAPHICS "F G".

```
620 LET TS=S
630 FOR X=0 TO 19: BORDER RND*7
 FOR U=0 TO 9: NEXT U: NEXT X
640 GO TO 420
650 FOR N=0 TO 2: BEEP .2,0: BE
EP .3,5: NEXT N: BEEP .3,2: BEEP
 .2,5: BEEP .3,6: BEEP .2,7: BEE
P .2,6: BEEP .2,5: BEEP .2,4
660 RETURN
670 RESTORE 680: FOR x=0 TO 55:
READ a: POKE USR "a"+x,a: NEXT
x: RUN
680 DATA 64,135,136,243,243,136
,135,64,2,225,17,207,207,17,225
2,74,0,181,0,146,0,170,73,254,19
2,160,144,136,128,128,0,127,3,5,
9,17,1,1,0,0,128,128,136,144,160
,192,254,0,1,1,17,9,5,3,127
```

VORTICE



```
10 REM VORTICE
       MAGNASCO,M. D. 1984
20 CLS : PRINT "  Este program
a toma una figura geometrica reg
ular, y la rota y disminuye sobr
e la pantalla."
50 INPUT "Numero de lados de l
a figura: ";N: INPUT "Angulo de
rotacion: ";F
70 CLS : PRINT N;AT 21,0;F: LE
T Q=2*PI/N: LET F=F*PI/360
80 IF N=3 THEN GO TO 200
100 FOR R=85 TO 0 STEP -2: LET
A=R*F
110 FOR I=0 TO N-1: LET xa=r*CO
S (a+i*Q): LET ya=r*SIN (a+i*Q):
 PLOT xa+128,ya+88: DRAW r*COS (
a+(i+1)*Q)-xa,R*SIN (a+(i+1)*Q)-
ya: NEXT I
120 NEXT R
130 STOP
200 FOR R=1 TO 85 STEP 2: LET A
=R*F: LET X1=R*COS (A+Q): LET Y1
=R*SIN (A+Q): LET X2=R*COS (A+Q+
Q): LET Y2=R*SIN (A+Q+Q): LET X3
=R*COS (A+Q+Q+Q): LET Y3=R*SIN (
A+Q+Q+Q): PLOT X1+128,Y1+88: DRA
W X2-X1,Y2-Y1: DRAW X3-X2,Y3-Y2:
 DRAW X1-X3,Y1-Y3: NEXT R
```

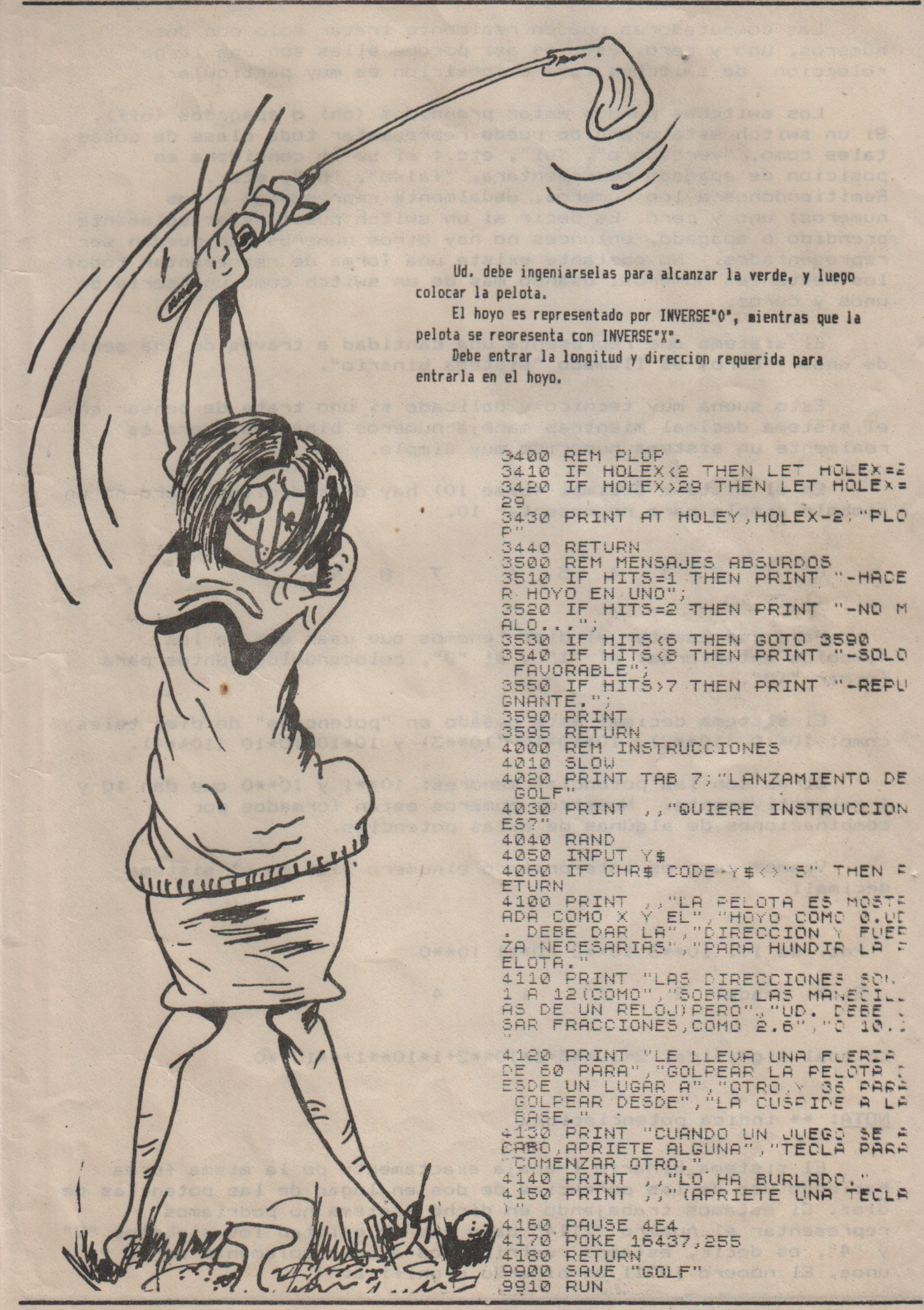
Es un programa que dibuja una figura geometrica regular, la rota y la disminuye de tamaño. Tiene dos partes: una parte para dibujar cualquier poligono, y otra especificamente para triangulos.

Ejemplo: 3, 8°




```
   1 REM *** GOLF***
  10 GOSUB 4000
  20 CLS
  30 REM DIBUJE BORDES
  40 FAST
  50 GOSUB 2000
 100 LET HOLEX=INT (RND*30)+1
 110 LET HOLEY=INT (RND*18)+1
 120 LET TEEX=INT (RND*30)+1
 130 LET TEEY=INT (RND*18)+1
 140 IF TEEX=HOLEX AND TEEY=HOLE
Y THEN GOTO 120
 150 LET HITS=0
 190 SLOW
 200 PRINT AT HOLEY,HOLEX;"0"
 210 PRINT AT TEEY,TEEX;"X"
 300 REM CURVA PPAL.
 310 GOSUB 3100
 320 PRINT "DIRECCION?"
 330 INPUT XD
 340 GOSUB 3000
 350 IF XD<0 OR XD>12 THEN GOTO
300
 360 PRINT TAB 22;XD
 370 GOSUB 3000
 380 PRINT "INTENSIDAD?"
 390 INPUT XS
 400 LET XS=INT XS
 410 GOSUB 3000
 420 PRINT TAB 28;XS
 500 LET A=TEEX*2
 510 LET B=INT ((TEEY*(-2))+42)
 520 LET XD=(XD*(-1))+15
 530 IF XD=12 THEN LET XD=0
 540 LET C=A+INT (COS ((PI/6)*XD
*XS)
 550 LET D=B+INT (SIN ((PI/6)*XD
*XS)
 560 LET XA=A
 570 LET XB=B
 580 GOSUB 1500
 590 LET P=1
 600 GOSUB 1000
 610 PRINT AT TEEY,TEEX;" ";
 620 LET TEEY=INT ((D-42)/(-2)+0
.5)
 630 LET TEEX=INT ((C+0.5)/2)
 640 LET HITS=HITS+1
 700 IF TEEX=HOLEX AND TEEY=HOLE
Y THEN GOTO 800
 710 LET A=XA
 720 LET B=XB
 730 LET P=0
 740 GOSUB 1000
 750 GOTO 200
 800 GOSUB 3400
 810 PRINT AT 20,0;"UD. LO HIZO"
 820 PRINT "LE LLEVO";HITS;"LANZ
AMIENTOS";
 830 GOSUB 3500
 840 IF INKEY$="" THEN GOTO 840
 850 CLEAR
 860 GOTO 20
1000 LET U=C-A
1010 LET V=D-B
1020 LET D1X=SGN U
1030 LET D1Y=SGN V
1040 LET D2X=SGN U
1050 LET D2Y=0
1060 LET M=ABS U
1070 LET N=ABS V
1080 IF M>N THEN GOTO 1130
1090 LET D2X=0
1100 LET D2Y=SGN V
1110 LET M=ABS V
1120 LET N=ABS U
1140 LET S=INT (M/2)
1150 FOR I=0 TO M
1160 IF P THEN PLOT A,B
1165 IF NOT P THEN UNPLOT A,B
1170 LET S=S+N
1180 IF S<M THEN GOTO 1230
1190 LET S=S-M
1200 LET A=A+D1X
1210 LET B=B+D1Y
1220 NEXT I
1230 LET A=A+D2X
1240 LET B=B+D2Y
1250 NEXT I
1260 RETURN
1500 REM PRUEBE NUEVA UBICACION
PARA TEE
1510 LET ERROR=(C<2 OR C>62 OR D
<6 OR D>42)
1520 IF NOT ERROR THEN RETURN
1530 PRINT AT 21,0;"DENTRO DE TE
RRENO ESCABROSO-GOLPE PENAL"
1540 LET HITS=HITS+1
1550 IF C<2 THEN LET C=2
1560 IF C>62 THEN LET C=62
1570 IF D<6 THEN LET D=6
1580 IF D>42 THEN LET D=42
1590 RETURN
2000 REM DIBUJE LIMITES VERDES
2010 LET Y=43
2020 GOSUB 2100
2030 LET Y=4
2040 GOSUB 2100
2050 LET X=0
2060 GOSUB 2200
2070 LET X=63
2080 GOSUB 2200
2090 RETURN
2100 REM DIBUJE LA LINEA
2110 FOR X=0 TO 63
2120 PLOT X,Y
2130 NEXT X
2140 RETURN
2200 REM DIBUJE LA LINEA"Y"
2210 FOR Y=4 TO 43
2220 PLOT X,Y
2230 NEXT Y
2240 RETURN
3000 REM LIMPIE APUNTES
3010 PRINT AT 20,0;"
                    "
3020 PRINT AT 20,0;
3030 RETURN
3100 REM LIMPIE LINEA 20
3110 PRINT AT 20,0;"

                    "
3120 PRINT AT 20,0;
3130 RETURN
```



Ud. debe ingeniarselas para alcanzar la verde, y luego colocar la pelota.

El hoyo es representado por INVERSE"0", mientras que la pelota se reoresenta con INVERSE"X".

Debe entrar la longitud y direccion requerida para entrarla en el hoyo.

```
3400 REM PLOP
3410 IF HOLEX<2 THEN LET HOLEX=2
3420 IF HOLEX>29 THEN LET HOLEX=
29
3430 PRINT AT HOLEY,HOLEX-2;"PLO
P"
3440 RETURN
3500 REM MENSAJES ABSURDOS
3510 IF HITS=1 THEN PRINT "-HACE
R HOYO EN UNO";
3520 IF HITS=2 THEN PRINT "-NO M
ALO...";
3530 IF HITS<6 THEN GOTO 3590
3540 IF HITS<8 THEN PRINT "-SOLO
 FAVORABLE";
3550 IF HITS>7 THEN PRINT "-REPU
GNANTE.";
3590 PRINT
3595 RETURN
4000 REM INSTRUCCIONES
4010 SLOW
4020 PRINT TAB 7;"LANZAMIENTO DE
 GOLF"
4030 PRINT ,,"QUIERE INSTRUCCION
ES?"
4040 RAND
4050 INPUT Y$
4060 IF CHR$ CODE Y$<>"S" THEN R
ETURN
4100 PRINT ,,"LA PELOTA ES MOSTR
ADA COMO X Y EL","HOYO COMO 0.UD
. DEBE DAR LA","DIRECCION Y FUER
ZA NECESARIAS","PARA HUNDIR LA P
ELOTA."
4110 PRINT "LAS DIRECCIONES SON
1 A 12(COMO","SOBRE LAS MANECILL
AS DE UN RELOJ)PERO","UD. DEBE U
SAR FRACCIONES,COMO 2.6","O 10.1
"
4120 PRINT "LE LLEVA UNA FUERZA
DE 60 PARA","GOLPEAR LA PELOTA D
ESDE UN LUGAR A","OTRO,Y 36 PARA
 GOLPEAR DESDE","LA CUSPIDE A LA
 BASE."
4130 PRINT "CUANDO UN JUEGO SE A
CABO,APRIETE ALGUNA","TECLA PARA
 COMENZAR OTRO."
4140 PRINT ,,"LO HA BURLADO."
4150 PRINT ,,"(APRIETE UNA TECLA
)"
4160 PAUSE 4E4
4170 POKE 16437,255
4180 RETURN
9900 SAVE "GOLF"
9910 RUN
```

Las computadoras pueden realmente tratar solo con dos numeros, uno y cero. Esto es asi porque ellas son una larga coleccion de switches cuya disposicion es muy particular.

Los switches pueden estar prendidos (on) o apagados (off). Si un switch esta prendido puede representar toda clase de cosas tales como, "verdadero", "si", etc.; si se lo considera en posicion de apagado representara, "falso", "no", etc.. Remitiendonos a los numeros, usualmente representan a dos numeros: uno y cero. Es decir si un switch puede estar solamente prendido o apagado, entonces no hay otros numeros que puedan ser representados. No obstante existe una forma de representar todos los restantes numeros, usando mas de un switch como una serie de unos y ceros.

El sistema que representa una cantidad a traves de una serie de unos y ceros es llamado "sistema binario".

Esto suena muy tecnico y delicado si uno trata de pensar en el sistema decimal mientras maneja numeros binarios, pero es realmente un sistema numerico muy simple.

En el sistema decimal (base 10) hay diez digitos, pero no un simbolo simple para representar 10.

0 1 2 3 4 5 6 7 8 9

Para representar el diez tenemos que usar dos de los simbolos anteriores: el "1" y el "0", colocandolos juntos para formar "10".

El sistema decimal esta basado en "potencias" de diez tales como: 10*10 (10**2), 10*10*10 (10**3) y 10*10*10*10 (10**4).

Otras son las potencias menores: 10**1 y 10**0 que dan 10 y 1 respectivamente. Nuestros numeros estan formados por combinaciones de algunas de estas potencias.

Veamos como es interpretado elnumero 2114 en el sistema decimal:

Pot. de 10:	10**3	10**2	10**1	10**0
Simb. usado:	2	1	1	4

Lo cual significa: 2*10**3+1*10**2+1*10**1+4*10**0.

<u>NOTA</u>: ** indica potenciacion.

El sistema binario trabaja exactamente de la misma forma haciendo uso de las potencias de dos en lugar de las potencias de diez. Si estamos trabajando en dicho sistema no podriamos representar el numero "2114" dado que no existen los simbolos "2" y "4", es decir, estamos restringidos a usar solamente ceros y unos. El numero 10011 en binario significa:

pot. de dos:	2**4	2**3	2**2	2**1	2**0
equiv. dec.:	16	8	4	2	1
simb. a usar:	1	0	0	1	1

cuyo significado es: 1*2**4+0*2**3+0*2**2+1*2**1+1*2* *0.

Si realizamos esta suma obtenemos el valor 19 en el sistema decimal.

El sistema binario presenta dificultades para trabajarlo, pues los numeros son representados con muchos digitos es muy facil equivocarse al escribirlos.

Cada uno de esos digitos binarios se denomina BIT (Binary digIT). Cada switch puede tener un bit, agrupandoselos generalmente de a ocho conformando un BYTE. Los 64 lotes de 1024 BYTES de memoria en el enteramente expandido TS1000 y ZX81 contiene

64*(1.024*8)=5.240.288 de estos switches.

Estos solo forman la "random acces memory", pero hay muchos mas de estos switches en otras partes de la computadora. Por que pensar en 1024?. Por que no 100, un numero mas simple?. La respuesta esta pregunta esta relacionada con la aritmetica binaria. Fundamentalmente, muchos los numeros que tienen sentido en computacion se relacionan con este sistema de numeracion.

Normalmente pensamos en nuestro familiar sistema decimal y en numeros tales como 100 o 10.000, como numeros puros y pues son potencias exactas de 10. En el sistema binario hay tambien numeros puros, pero expresados en potencias exactas de 2 y no de 10.

Entonces, para la computadora, un numero exacto es 1024 y no 1000, ya que el primero es potencia de 2 y el segundo de 10.

La razon por la cual un byte (8 bits) no puede almacenar mas que 255, es que si los ocho switches estan prendidos, el valor representado es:

Nro. bit:	8	7	6	5	4	3	2	1
Pot. dos:	2**7	2**6	2**5	2**4	2**3	2**2	2**1	2**0
valor :	128	64	32	16	8	4	2	1

cuya suma es el valor 255 en decimal. Sumando un unoa este valor resultara un "acarreo" tal como cuando al sumar 1 a 9.999 genera un acarreo.

```
  11111111        9.999
+        1      +     1
 _________      _______
 100000000       10.000
```

En ambos casos el acarreo significa que hay un digito extra en el resultado. Un byte de 8 bits no puede contener un noveno bit, este bit extra provoca lo que se denomina "overflow".

Usando dos bytes por numero, podemos tener 65.336 si todos los switches estan en on.

1er. byte: 11111111
2do. byte: 11111111

Pero el decimosexto bit tiene un especial proposito: representar el signo mas o el menos. Solamente siete de los digitos pueden ser usados para representar el numero y entonces, solamente 32.767 puede ser almacenado. Esto muestra una seria reduccion pero no implica un desperdicio del sistema. Antes, el rango de numeros almacenables iba de 0 a 65.536 y ahora, usando el octavo bit del segundo byte como bit de signo, el rango va desde -32.767, a traves de 0, hasta 32.766.

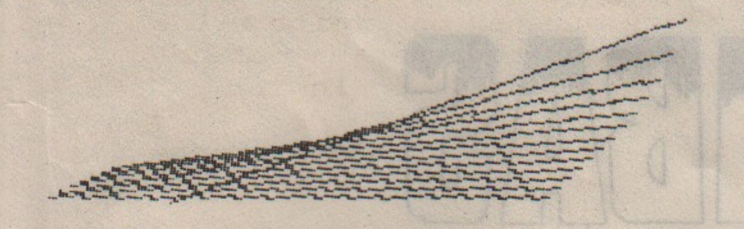
64K es el numero maximo de direcciones de memoria que el procesador Z80 puede tratar. Nuestras computadoras tienen un set de caracteres con 255 codigos en el. Ese rango de 0 a 255 da un total de 256 numeros, numero que es potencia de dos. Guardar uno cualquiera de tales numeros ocupa tan solo un byte.

Las cadenas de caracteres (strings) pueden tener, como maximo, 256 caracteres, por las mismas razones.

3D. graF C.C graF

TS 2068/SPECTRUM

```
   1 REM    CC-GRAF
        MAGNASCO, M O 1984
   5 IF PEEK 23746<>0 THEN GO TO
 100
  10 CLEAR 63205: LET u=63206: R
ESTORE : READ a,b,c,d,e,f: DATA
10,11,12,13,14,15
  20 LET u$="*2100603600237cfe78
20f8c9f33e01d3f4dbffcbffd3ff3e80
f5fbcd8e0ef3dbffcbbfd3ffafd3f4f1
fe80200332c25cfbc9"
  30 FOR n=2 TO LEN u$-1 STEP 2:
 POKE u,16*VAL u$(n)+VAL u$(n+1)
: LET u=u+1: NEXT n
  40 RANDOMIZE USR 63218
  50 FOR a=63256 TO 63263: POKE
a,248: POKE a+8,255-PEEK (a+16):
 POKE a+16,255: NEXT a
 100 CLS : INPUT "ENTRE F(x) [0<
=f<=7.4]          F(x,y)=";f$: I
NPUT "Escala en x";ex'"Escala en
 y";ey: LET ex=ex/16: LET ey=ey/
128
 105 OUT 255,2: LET u=24576: GO
SUB 300
 110 FOR z=0 TO 2: FOR j=0 TO 7:
 FOR i=j+64*z TO j+64*z+56 STEP
8: FOR k=0 TO 31: LET x=ex*(k-15
)
 120 LET y=(i-96)*ey: LET in=INT
 VAL f$: LET x=x+ex/2: LET pa=8*
INT VAL f$: POKE u,pa+in: LET u=
u+1: NEXT k: NEXT i: NEXT j: NEX
T z
 200 BEEP .1,50: PAUSE 0: OUT 25
5,0
 250 STOP
 300 FOR i=0 TO 21: PRINT AT i,0
;"██████████████████████████████
██": NEXT i: RETURN
 600 FOR i=24576 TO 30719: POKE
i,0: NEXT i: RETURN
```



Son dos programas hermanos destinados a un mismo fin: graficar funciones de dos variables.

3D da una representacion tridimensional de la funcion (en perspectiva caballera). Sirve primoldialmente para graficar superficies de variacion suave, como polinomios, gaussianas, etc..

Ejemplo:

.
F(x,y)=EXP(-x*x-y*y)
. rango de x=2
. rango de y=2

NOTA: en todo lugar de un programa donde haya que evaluar x↑2, A↑3, conviene escribir x*x, A*A*A. El motivo, es que para evaluar a una potencia, la maquina ejecuta:

. A↑B=EXP(B*Ln(A)),

cuya evaluacion es mucha mas lenta.

CC da una representacion codificada en color (estilo fotografia sintetica o centellograma) del valor de F(x,y). Debido a que solo hay disponibles 8 colores, este programa no da demasiada informacion sobre gaussianas o polinomios. Sirve mas para curvas de variacion rapida cuya representacion en 3D seria confusa.

Ejemplo:

. F(x,y)=3.6*(1+(COS(x*x-y*y))
variacion de x=5
variacion de y=5

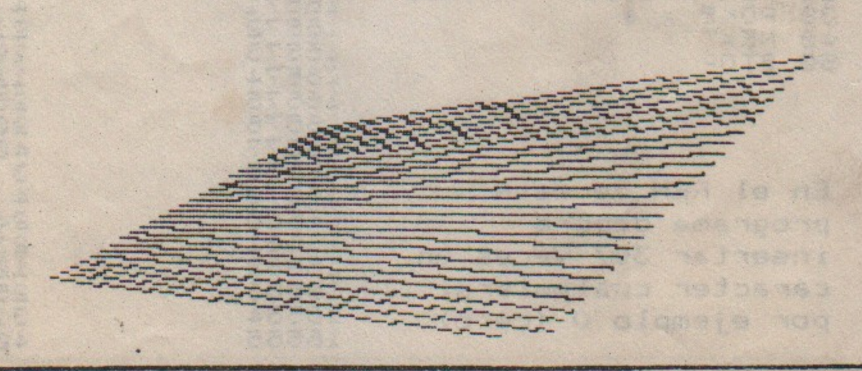
Para graficar una familia de curvas U(x,y)=cte.,

F(x,y)=3.6*(1+COS(U(x,y))

donde U(x,y)=x*x+y*y
=x*x-y*y
=x*x
=x*x-y*y+x*y

```
  10 REM 3D GRAF
       MAGNASCO, M.O. 1984
  20 PAPER 0: INK 7: CLS : PRINT
 '" Este es un programa destina
do agraficar funciones de dos va
ria-bles."'" Ud debera ingresar
la funcion, (que variara solo en
tre 0 y 1)  y dos parametros de
escala, que define-n cual es el r
ango que    tomaran x e y."'
  30 OVER 1: INK 4: PLOT 96,16:
DRAW 40,0: DRAW 0,40: DRAW -40,0
: DRAW 0,-40: PLOT 90,36: DRAW 8
5,0: DRAW -4,2: PLOT 116,8: DRAW
 0,80: DRAW -2,-4
  35 PRINT AT 15,12; PAPER 1;"
  ";AT 16,12;"     ";AT 17,12;"
     ";AT 18,12;"     ";AT 19,12
;"     "
  40 INK 7: PRINT AT 16,9;"-ex";
AT 16,18;"ex";AT 13,15;"ey";AT 2
0,15;"-ey"
  50 INPUT " Entre f(x,y)    [ 0<
=f<=1 ]       f(x,y)=";f$
  60 INPUT "Escala en x ";ex'"Es
cala en y ";ey: LET ez=85: LET e
x=ex/75: LET ey=ey/75
 100 OVER 0: CLS

 110 FOR j=150 TO 0 STEP -5: LET
 xa=j*.6: LET ya=j*.6: INVERSE 1
: FOR i=0 TO 150 STEP 5
 120 LET x=(i-75)*ex: LET y=(j-7
5)*ey: LET z=VAL f$: LET xp=i+.6
*j: LET yp=z*ez+.6*j: PLOT xa,ya
: DRAW xp-xa,yp-ya: INVERSE 0: L
ET xa=xp: LET ya=yp: NEXT i: PLO
T xp,yp: NEXT j
 130 PAUSE 0: GO TO 60
```



BOMBAS

Usted posee un avion cargado con bombas y ellas seran utilizadas para destruir edificios. Los edificios se destruyen por completo si las bombas lo tocan en la cima; si caen mas abajo el pedazo de arriba no queda destruido por lo que debera dispararle de nuevo. Pero debe cuidarse de no malgastarlas pues las que posee son pocas. El avion parte de la linea uno y bajara a la siguiente cada vez que termine de recorrer una. Para no estrellarlo contra un edificio Ud. tratara de destruir primero los mas altos. El juego termina cuando logra destruir todos los edificios y aterrizar el avion. Para entrar el codigo de maquina tipee primero el siguiente programa cargador:

```
5 REM
10 INPUT a
20 FOR x=16514 TO 16627
30 POKE x,a
40 NEXT x
50 STOP
```

En el REM de este programa debera insertar 307 veces un caracter cualquiera, por ejemplo 0 (cero).

16514	128	16586	3
16515	151	16587	2
16516	146	16588	24
16517	33	16589	210
16518	33	16590	[illegible]
16519	169	16591	0
16520	1	16592	6
16521	0	16593	197
16522	7	16594	205
16523	205	16595	245
16524	245	16596	8
16525	8	16597	17
16526	17	16598	130
16527	192	16599	64
16528	1	16600	1
16529	62	16601	1
16530	128	16602	2
16531	215	16603	205
16532	[illegible]	16604	10
16533	122	16605	11
16534	179	16606	42
16535	32	16607	16
16536	248	16608	64
16537	[illegible]	16609	17
16538	3	16610	33
16539	[illegible]	16611	0
16540	138	16612	43
16541	215	16613	229
16542	[illegible]	16614	126
16543	253	16615	254
16544	[illegible]	16616	169
16545	[illegible]	16617	32
16546	251	16618	14
16547	64	16619	35
16548	42	16620	126
16549	50	16621	254
16550	64	16622	118
16551	[illegible]	16623	40
16552	34	16624	250
16553	50	16625	254
16554	64	16626	128
16555	124	16627	40
16556	230	16628	41
16557	[illegible]	16629	225
16558	198	16630	195
16559	[illegible]	16631	141
16560	[illegible]	16632	5
16561	197	16633	254
16562	205	16634	146
16563	245	16635	32
16564	8	16636	[illegible]
16565	[illegible]	16637	[illegible]
16566	50	16638	128
16567	64	16639	35
16568	230	16640	126
16569	[illegible]	16641	254
16570	198	16642	118
16571	4	16643	40
16572	215	16644	250
16573	193	16645	254
16574	120	16646	138
16575	254	16647	40
16576	20	16648	18
16577	40	16649	25
16578	3	16650	126
16579	4	16651	254
16580	24	16652	128
16581	235	16653	40
16582	121	16654	10
16583	254	16655	254
16584	31	16656	138
16585	40		

16657	40	16740	254	16784	[illegible]
16658	8	16741	25	16785	[illegible]
16659	54	16742	32	16786	[illegible]
16660	128	16743	1	16787	62
16661	25	16744	32	16788	151
16662	126	16745	[illegible]	16789	215
16663	24	16746	30	16790	16
16664	246	16747	60	16791	253
16665	54	16748	64	16792	62
16666	146	16749	193	16793	33
16667	33	16750	12	16794	50
16668	134	16751	121	16795	133
16669	64	16752	254	16796	4
16670	225	16753	25	16797	50
16671	[illegible]	16754	32	16798	134
16672	4	16755	[illegible]	16799	64
16673	11	16756	120	16800	201
16674	120	16757	254	16801	28
16675	17	16758	20	16802	28
16676	32	16759	40	16803	28
16677	251	16760	31	16804	28
16678	58	16761	24	16805	28
16679	62	16762	[illegible]	16806	28
16680	64	16763	254	16807	28
16681	60	16764	32	16808	28
16682	50	16765	32	16809	28
16683	2	16766	3	16810	28
16684	64	16767	14	16811	28
16685	203	16768	0	16812	28
16686	11	16769	4	16813	28
16687	32	16770	121	16814	28
16688	173	16771	230	16815	28
16689	58	16772	1	16816	28
16690	60	16773	198	16817	28
16691	64	16774	150	16818	28
16692	61	16775	50	16819	28
16693	40	16776	131	16820	28
16694	54	16777	64	16821	28
16695	58	16778	195	16822	28
16696	11	16779	209	16823	28
16697	64	16780	64	16824	28
16698	60	16781	193	16825	28
16699	40	16782	205	16826	28
16700	48	16783	245	16827	28
16701	25				
16702	126				
16703	254				
16704	128				
16705	40				
16706	10				
16707	254				
16708	138				
16709	40				
16710	8				
16711	54				
16712	128				
16713	25				
16714	126				
16715	24				
16716	246				
16717	54				
16718	146				
16719	33				
16720	134				
16721	64				
16722	126				
16723	254				
16724	28				
16725	3				
16726	3				
16727	4				
16728	11				
16729	245				
16730	254				
16731	246				
16732	61				
16733	119				
16734	254				
16735	28				
16736	32				
16737	11				
16738	43				
16739	126				

1 BYTE

Todos los caracteres graficos desde el teclado se usan dentro de strings. Todas las palabras que se encuentran en el teclado y los simbolos, sin importar cuantos caracteres contengan, son entradas entre comillas.

Los corchetes y los simbolos matematicos tales como "*", "+", "/", "-" y "**" ocupan un byte cada uno. Todos los signos de puntuacion cuestan un byte pero la coma cuando es usada como un separador de impresion, cuesta 15 bytes de la memoria dedicada a la pantalla en el archivo de display.

NOT, usado en una instruccion condicional ocupa un byte.

6 BYTES

Todas las siguientes lineas de programa usan 6 bytes cada una:

```
10 CLS        10 CLEAR
10 CONT       10 COPY
10 FAST       10 SLOW
10 LLIST      10 LIST
10 LPRINT     10 PRINT
10 REM        10 RAND
10 RETURN     10 SCROLL
10 STOP
```

7 BYTES

Estas son versiones principalmente de lineas de 6 bytes las cuales pueden aparacer con caracteres extras. La linea que imprime PI a pesar que es una funcion que se representa en dos caracteres ocupa solamente un byte de programa.

```
10 INPUT A     10 REM A
10 PRINT A     10 LPRINT A
10 PRINT PI    10 PRINT RDN
```

La ultima linea es usada frecuentemente con otras funciones y sera tratada con mas cuidado en la seccion que se refiere a 18 BYTES.

8 BYTES

```
10 PRINT A$    10 INPUT A$
10 PRINT ""    10 LET A=B
10 PRINT CHR$ A
10 PRINT STR$ A
```

Esta ultima linea muestra que tenemos un costo adicional de un byte para la funcion CHR$.

9 BYTES

```
10 PRINT LEN A$      10 PRINT VAL A$
10 LPRINT "A"        10 PRINT CODE A$
10 LET A=B           10 PRINT "*"
10 PRINT A
```

Por esto podemos ver que LEN, VAL, CODE ocupan un byte cada una.

10 BYTES

```
10 LET A$=INKEY$
AND B=1 ⎫
        ⎬ como parte de IF/THEN
OR B>=1 ⎭
```

11 BYTES

```
10 LET A$="" (string vacio)
10 PRINT "A";
```

Se puede ver que ";" cuesta un byte

12 BYTES

```
10 LET A$="A"
TAB (ver 20 BYTES)
```

13 BYTES

```
10 PAUSE
```

pero,

```
10 PAUSE 10   cuesta 14 bytes
10 PAUSE 100  cuesta 15 bytes
10 PAUSE 1000 cuesta 16 bytes
```

```
10 GO TO 1
```

pero,

```
10 GO TO 10  cuesta 14 bytes
10 GO TO 100 cuesta 15 bytes
```

14 BYTES

```
10 PRINT STR$ 1
```

Este punto es importante para señalar un interesante factor. Nuestras computadoras permitiran uso de nombres de variables n las lineas programas, siendo que otras computadoras insisten en conocer primero el valor. Los usuarios de TS 1000 pueden indicar GO TO A o GO SUB X y la computadora obedecera el comando sin importarle cuando conocera el valor de esas variables. Bajo el encabezamiento "8 BYTES" hay un ejemplo similar. La diferencia es que esa linea tiene un nombre de variable y no un numero. El ahorro de memoria es de 6 bytes en 14 bytes, 43%. Esto es de veras valioso y su uso se vera cuando por diferentes circunstancias los costos de memoria sean investigados.

15 BYTES

```
10 LET A=1
```

pero,

```
10 LET A=10    cuesta 16 bytes
10 LET A=100   cuesta 17 bytes
10 LET A=1.1   cuesta 17 bytes


10 LET A=B     cuesta 9 bytes.
```

Entonces el beneficio de usar nombres de variables es mejor que usar numeros en un 40%.

16 BYTES

```
10 LET A=SIN 1
```

El costo es el mismo para las funciones: COS, TAN, ASN, ACS, ATN, INT, SGN, ABS, SQR, LN y EXP. Usando nombres de variables tambien se reducen los costos.

```
10 LET A= SIN A  usa 10 bytes
10 DIM A(1)      usa 16 bytes pero,
10 DIM A(10)     usa 17 bytes
10 DIM A(100)    usa 18 bytes
10 DIM A(1,1)    usa 24 bytes de los cuales 8 bytes son para
```

" ,1".

Compare esto con lo siguiente:

```
10 DIM (A)    usa 10 bytes
10 DIM (A,B)  usa 12 bytes
```

Cuando se trabaja en un programa con arreglos permite ocupar cinco bytes por numero si se trata de dimensiones simples. En arreglos multidimensionales se realiza el producto de los numeros entre parentesis y al resultado se lo multiplica por cinco. Esta cantidad es el numero de bytes que ocupara el arreglo en la zona de almacenamiento de variables.

```
10 DIM A(5,10) ocupa 256 bytes para el arreglo A
```

```
10 PEEK (1)
```

pero,

```
10 PRINT PEEK (10)  usa 17 bytes
10 PRINT PEEK (100) usa 18 bytes
```

El tamaño de la direccion mas usual que aparece en un PEEK es de la forma:

```
10 PRINT PEEK (10000) y usa 20 bytes.
```

17 BYTES

10 DIM A$(1) usa un byte mas que
10 DIM A(1) entonces todos los arreglos de tipo caracter usan byte mas que los arreglos numericos. Hay una diferencia importante entre los dos tipos de arreglos, los arreglos string solo ocupan un byte por caracter y no cinco.

```
10 DIM A$(5,10) ocupa 50 bytes
```

18 BYTES

```
10 PRINT INT (RND*9)
```

Esta es la forma mas usual de usar la funcion RND. Aca tambien si se usan nombres de variables se reduce el costo.

```
10 PRINT INT (RND X) usa solamente 12 bytes.
```

```
10 PRINT TAB 1;"A"
```

pero la cantidad de bytes que usa TAB depende de la posicion donde se quiera imprimir. El costo afecta al archivo del display y no al area de memoria donde se almacenan las lineas de programa.

```
10 PRINT TAB X;"A" usa solo 12 bytes.
```

Como el costo de:

10 PRINT "A" es 9 bytes, el de "TAB X;" sera solamente de 3 bytes.

19 BYTES

El costo de subrutinas es de 19 bytes como minimo.

```
1 GO TO 9  cuesta 13 bytes
9 GO SUB 5
5 RETURN
```

todo esto cuesta 32 bytes por lo tanto la subrutina ocupa 19 bytes.

21 BYTES

```
10 POKE 1,1
```

pero,

```
10 POKE 10,1       cuesta 22 bytes
10 POKE 10000,1    cuesta 25 bytes
10 POKE 10000,10 cuesta 26 bytes
10 POKE A,A cuesta 9 bytes
10 PLOT 1,1
```

pero,

```
10 PLOT 1,10   cuesta 22 bytes
10 PLOT 1,40   cuesta 23 bytes
10 PLOT A,A    cuesta 9 bytes
```

Todos los comandos UNPLOT tienen el mismo costo. Una o dos posiciones de ploteo le ocupan 1 byte al archivo de display por cada posicion impresa.

23 BYTES

```
10 LET A=2*2 cuestan 23 bytes pero como
10 LET A=2 cuesta 15
```

el costo del numero y el signo de multiplicacion es de 8 bytes. Un metodo alternativo para calcular raices cuadradas cuesta la misma cantidad de memoria:

```
10 LET A=2**2
10 LET A=-2*-2 cuesta 25 bytes y
10 LET A=-2**2 se reduce a 24 bytes pero da respuesta
```
incorrecta!!

Todos los numeros al cuadrado son positivos y la computadora da resultado negativo. Esto significa una falla en el lenguaje.

Si usted esta escribiendo un programa donde calcula potencias cuadradas le sera necesario incluir la funcion ABS en todas las lineas que incluyen a funcion "**".

10 FOR J=1 TO 9 cuesta 23 bytes pero,
10 FOR J=1 TO 10 cuesta 24 bytes
10 FOR J=1 TO 9 STEP 2 cuesta 31 bytes pues "STEP 2" cuesta 8 bytes.

El costo basico del loop FOR/NEXT es:

```
10 FOR J=1 TO 9  ⎫
                 ⎬ 30 bytes
20 NEXT J        ⎭
```

24 BYTES

10 IF A=1 THEN GO TO 9 pero
10 IF A=1 THEN GO TO 10 cuesta 25 bytes.

El uso de "<=", "<", ">" y "<>" en cualquier linea cuestan siempre los mismo.

10 IF A=1 OR B<2 THEN GO TO 9 cuesta 34 bytes pues "OR B<2" cuesta 10.

10 IF NOT A=1 THEN GO TO 9 cuesta 25 bytes pues "NOT" ocupa tan solo 1 byte.

10 IF A=1 THEN GO SUB 9 tambien cuesta 24 bytes

pero ella incluira al menos una linea extra GO TO para saltar esta subrutina. Si todas las subrutinas estan al final del programa la sentencia STOP hara que se reduzca su costo en unos 6 bytes.

Nuestras computadoras aceptaran nombres de variables tanto en el enunciado GO TO como en el GO SUB y seran guardadas como las tiene.

IMPRIMIENDO CON O SIN FORMATOS

PRINT AT

10 PRINT "A" cuesta basicamente 9 bytes

10 PRINT AT 1,1;"A" cuesta 26 bytes pues "AT 1,1;" ocupa 17, la mayoria de estos para los dos numeros.

10 PRINT AT L,L;"A" cuesta solamente 14 bytes

Cada posicion de impresion a lo largo de una linea cuesta un byte extra al archivo del display.

TAB

10 PRINT TAB 1;"A" cuesta 18 bytes pues "TAB 1;" ocupa 9 bytes

10 PRINT TAB X;"A" cuesta solamente 12 bytes del espacio para programa.

COMPARANDO EL COSTO DEL PRINT AT CON EL PRINT VACIO

10 PRINT AT 4,1;"A" cuesta 26 bytes pero,

```
10 PRINT          )
11 PRINT          )  cuesta
12 PRINT          ) 27 bytes
13 PRINT "A"      )
```

El precio de usar enunciados PRINT vacios para espaciar un texto es barato cuando se quiere separar una o dos lineas. Para dejar tres o mas lineas blanco es mas conveniente usar PRINT AT pues es de menor costo.

```
10 PRINT
                  cuesta 24 bytes
11 PRINT TAB 5;"A"
```

mientras que:

10 PRINT AT 2,5;"A" cuesta 26 bytes.

10 PRINT AT X,Y;"A" cuesta solamente 14 bytes para almacenarlo en el area de programa.

STRING

10 PRINT A$ cuesta basicamente 8 bytes
10 PRINT A$(TO 9) cuesta 18 bytes pues "(TO 9)" ocupa 10 bytes
10 PRINT A$(TO 10) cuesta 9 bytes pues hay un caracter mas
10 PRINT A$(1 TO 9) ocupan 25 bytes de los cuales 17 son el costo de "(1 TO 9)"

La ultima alternativa:

10 PRINT A$(1 TO) cuesta 18 bytes tal como Ud. se lo imaginaba

10 PRINT A$(A TO B) cuesta 13 bytes

comparados con los 25 de la linea que usa directamente valores.

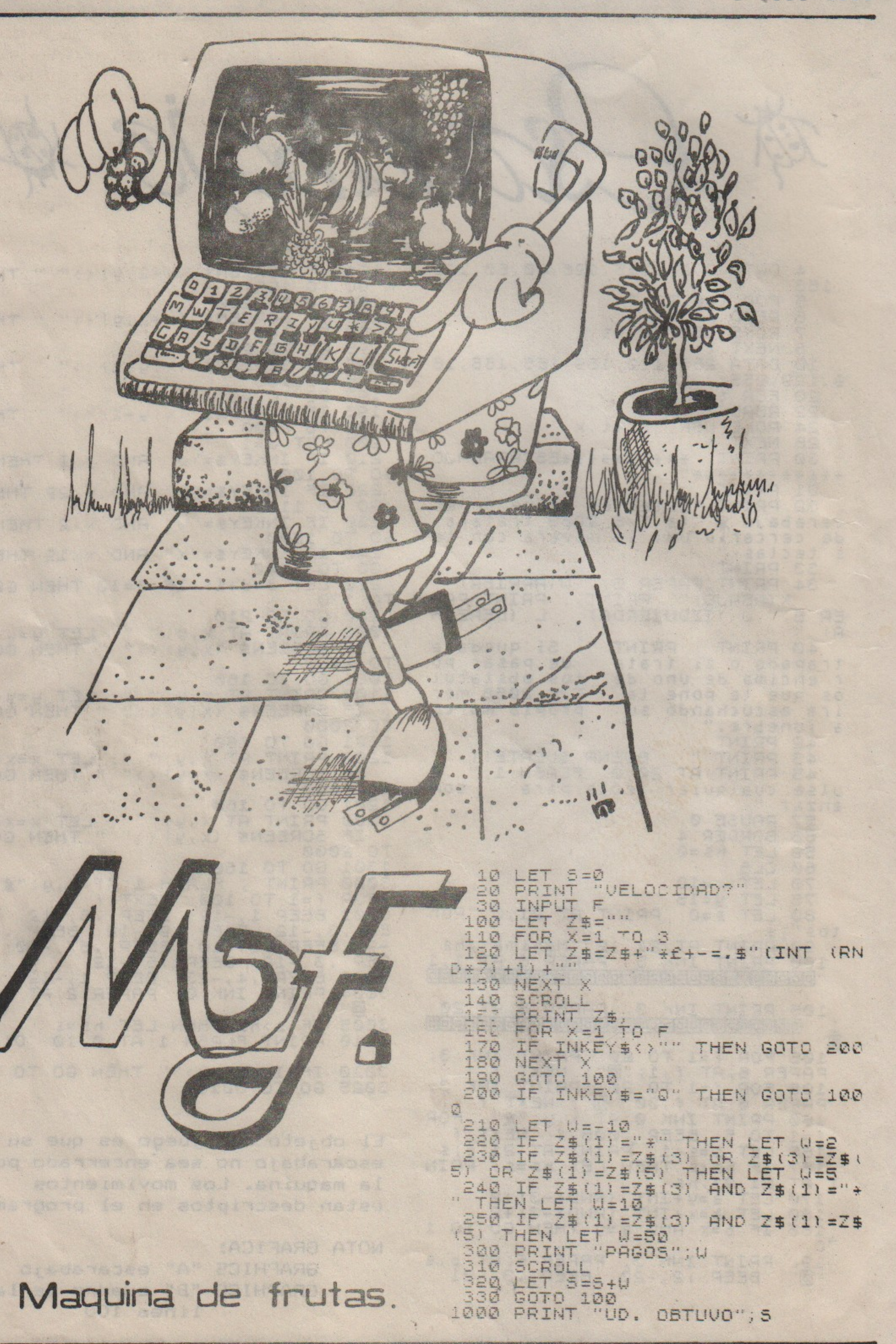
Escarabajo

```
   4 DATA 0,24,165,126,60,60,126
,153
   5 FOR t=0 TO 7
   6 READ x
   7 POKE USR "a"+t,x
   8 NEXT t
  10 DATA 255,129,189,165,165,18
9,129,255
  20 FOR t=0 TO 7
  22 READ X
  24 POKE USR "b"+t,x
  26 NEXT t
  30 PRINT "**********ESCARABAJO
************"
  31 PRINT
  32 PRINT "Usted es ahora un es
carabajo ✱ la T/S 2068 tratara
de cercarlo ud. se movera con la
s teclas "
  33 PRINT
  34 PRINT PAPER 6;" W(ARRIBA)
   X(ABAJO)": PRINT : PRINT PAP
ER 6;" J (IZQUIERDA)   L (DERECH
A) "
  40 PRINT : PRINT "  Si queda a
trapado o si trata  de pasar po
r encima de uno de  los obstacul
os que le pone la   T/s 2068 mor
ira escuchando su   propia march
a funebre."
  42 PRINT
  43 PRINT "   BUENA SUERTE!! "
  45 PRINT AT 20,0; FLASH 1;"  P
ulse cualquier tecla para    com
enzar "
  57 PAUSE 0
  58 BORDER 4
  59 LET hs=0
  60 CLS
  70 LET x=10
  75 LET y=15
  80 LET s=0: PRINT AT 21,0;"Pun
tos";s
  90 PRINT AT 21,12;"Maxima";hs
 100 PRINT INK 3; PAPER 6;AT 1,1
 "▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣"

 105 PRINT INK 3; PAPER 6;AT 20,
1 "▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣
▣"
 106 FOR f=1 TO 20: PRINT INK 3;
 PAPER 6;AT f,1;"▣": NEXT f
 108 FOR f=1 TO 20: PRINT INK 3;
 PAPER 6;AT f,30;"▣": NEXT f
 160 PRINT INK 0;AT x,y;"✱": FOR
 f=1 TO 5: BEEP .005,f: NEXT f
 165 LET s=s+1: PRINT AT 21,6;s
 167 IF hs<s THEN LET hs=s: PRIN
T AT 21,21;hs
 170 LET a=y+INT (RND*-3)+2
 180 LET b=x+INT (RND*-3)+2
 185 IF b=x AND a=y THEN GO TO 1
70
 190 PRINT INK 3; PAPER 6;AT b,a
 "▣": BEEP .2,-20: BEEP .3,-21
 191 IF SCREEN$ (x-1,y)<>" " THE
N GO TO 193
 192 GO TO 200
 193 IF SCREEN$ (x+1,y)<>" " THE
N GO TO 196
 195 GO TO 200
 196 IF SCREEN$ (x,y+1)<>" " THE
N GO TO 198
 197 GO TO 200
 198 IF SCREEN$ (x,y-1)<>" " THE
N GO TO 3000
 200 LET d=1
 210 IF INKEY$="J" AND _>2 THEN
GO TO 1000
 220 IF INKEY$="L" AND y<29 THEN
 GO TO 1100
 230 IF INKEY$="W" AND x>2 THEN
GO TO 1200
 240 IF INKEY$="X" AND x<19 THEN
 GO TO 1300
 250 LET d=d+1: IF d=10 THEN GO
TO 170
 260 GO TO 210
1000 PRINT AT x,y;" ": LET y=y-1
: IF SCREEN$ (x,y)<>" " THEN GO
TO 2000
1001 GO TO 160
1100 PRINT AT x,y;" ": LET y=y+1
: IF SCREEN$ (x,y)<>" " THEN GO
TO 2000
1101 GO TO 160
1200 PRINT AT x,y;" ": LET x=x-1
: IF SCREEN$ (x,y)<>" " THEN GO
TO 2000
1201 GO TO 160
1300 PRINT AT x,y;" ": LET x=x+1
: IF SCREEN$ (x,y)<>" " THEN GO
TO 2000
1301 GO TO 160
3000 PRINT ; FLASH 1;AT x,y;"✱"
 FOR f=1 TO 100: NEXT f
3001 BEEP 1,-12: BEEP .6,-12: BE
EP .3,-12: BEEP .9,-12: BEEP .6,
-9: BEEP .3,-10: BEEP .57,-10: B
EEP .3,-12: BEEP .6,-12
3002 BEEP .4,-13: BEEP 1,-12
3004 PRINT INK 0; PAPER 2;AT x,y
 "▣"
3005 IF s>hs THEN LET hs=s
3010 PRINT FLASH 1;AT 0,10 "Otra
?"
3020 IF INKEY$<>"" THEN GO TO 60
3025 GO TO 3015
```

El objeto del juego es que su escarabajo no sea encerrado por la maquina. Los movimientos estan descriptos en el programa.

NOTA GRAFICA:
GRAPHICS "A" escarabajo
GRAPHICS "B" cuadro de la linea 100



Maquina de frutas.

```
  10 LET S=0
  20 PRINT "VELOCIDAD?"
  30 INPUT F
 100 LET Z$=""
 110 FOR X=1 TO 3
 120 LET Z$=Z$+"*£+-=.$"(INT (RN
D*7)+1)+""
 130 NEXT X
 140 SCROLL
 150 PRINT Z$,
 160 FOR X=1 TO F
 170 IF INKEY$<>"" THEN GOTO 200
 180 NEXT X
 190 GOTO 100
 200 IF INKEY$="0" THEN GOTO 100
0
 210 LET W=-10
 220 IF Z$(1)="*" THEN LET W=2
 230 IF Z$(1)=Z$(3) OR Z$(3)=Z$(
5) OR Z$(1)=Z$(5) THEN LET W=5
 240 IF Z$(1)=Z$(3) AND Z$(1)="*
" THEN LET W=10
 250 IF Z$(1)=Z$(3) AND Z$(1)=Z$
(5) THEN LET W=50
 300 PRINT "PAGOS";W
 310 SCROLL
 320 LET S=S+W
 330 GOTO 100
1000 PRINT "UD. OBTUVO";S
```

```
   1 REM 16K MAQUINA DE FRUTA
   2 REM ********************
  20 PRINT "COMIENZO VELOCIDAD?"
  35 IF F<1 THEN GOTO 20
  40 PRINT "CUANTO TIEMPO? (MINU
TOS)"
  50 INPUT N
  60 IF N<1 OR N>10 THEN GOTO 40
  70 POKE 16437,128+N*12
 105 IF PEEK 16437=128 THEN GOTO
 2000
 325 LET F=F-(W/10)
 327 IF F<1 THEN LET F=1
1010 STOP
2000 SCROLL
2010 PRINT "***TIEMPO ARRIBA***"
2020 SCROLL
2030 GOTO 1000
```

El programa le mostrara continuamente una seleccion de tres caracteres, y Ud. podra presionar una tecla cuando sea impresa una combinacion ganadora. Si es lo suficientemente rapido su puntaje se incrementara rapidamente, ya que el puntaje depende de cuantas combinaciones haya sido capaz de parar. Las combinaciones ganadoras son:

*?? paga 2

**? paga 10

cualquiera que tenga 2 caracteres iguales paga 5 (excepto **?)

Cualquiera que tenga 3 caracteres paga 50

Inicialmente puede elegir la velocidad para seleccionar. Presione "Q" para hacer su score. Si Ud. presiona "Q" cuando esta seleccionando perdera 10 puntos.

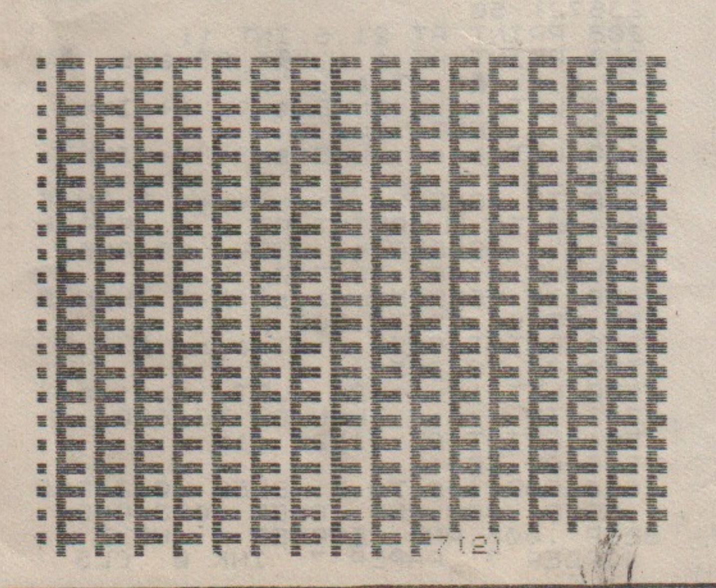
16 K: si posee expansion a 16 K agregue este programa.

Patrones Universales

Este programa produce patrones graficos al azar. Hay mas de dos millones de disenos que pueden verse en pantalla o imprimirse. Cada patron tiene un numero y una longitud de string impresas en el fondo. Con esos datos puede ser llamado nuevamente. Responda que NO cuando le pregunta por patrones al azar y entre esos numeros.

4(12)

```
 10 DIM A(33)
 20 PRINT "MODELO AL AZAR? (S/N
 30 INPUT A$
 40 CLS
 50 IF NOT A$="S" THEN GOTO 200
 60 LET L=INT (RND*33)+1
 70 LET X=INT (RND*65535)+1
 80 RAND X
 85 FAST
 90 FOR J=1 TO L
100 LET A(J)=INT (RND*10)+1
110 NEXT J
120 FOR J=1 TO 660/L
130 FOR K=1 TO L
140 PRINT CHR$ A(K);
150 NEXT K
160 NEXT J
170 PRINT X;"(";L;")"
180 STOP
200 PRINT "NUMERO MODELO?"
210 INPUT X
220 CLS
230 PRINT "LONGITUD DEL CORDON?
240 INPUT L
250 CLS
260 GOTO 80
```



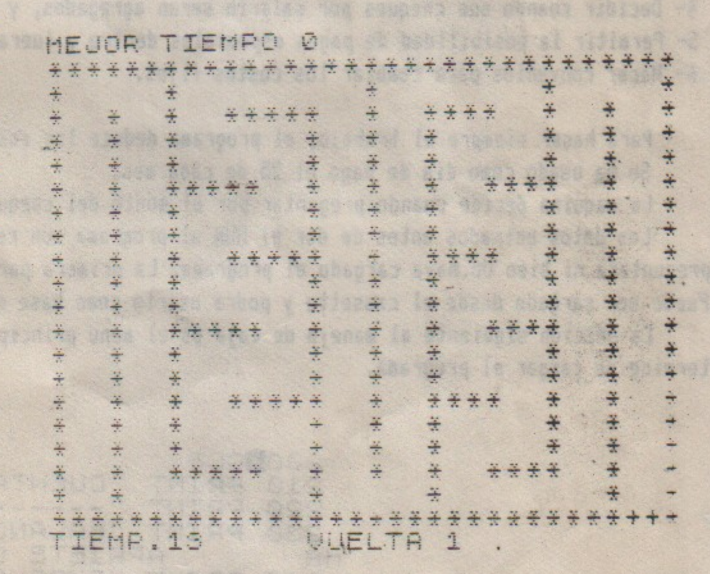
CARRERA

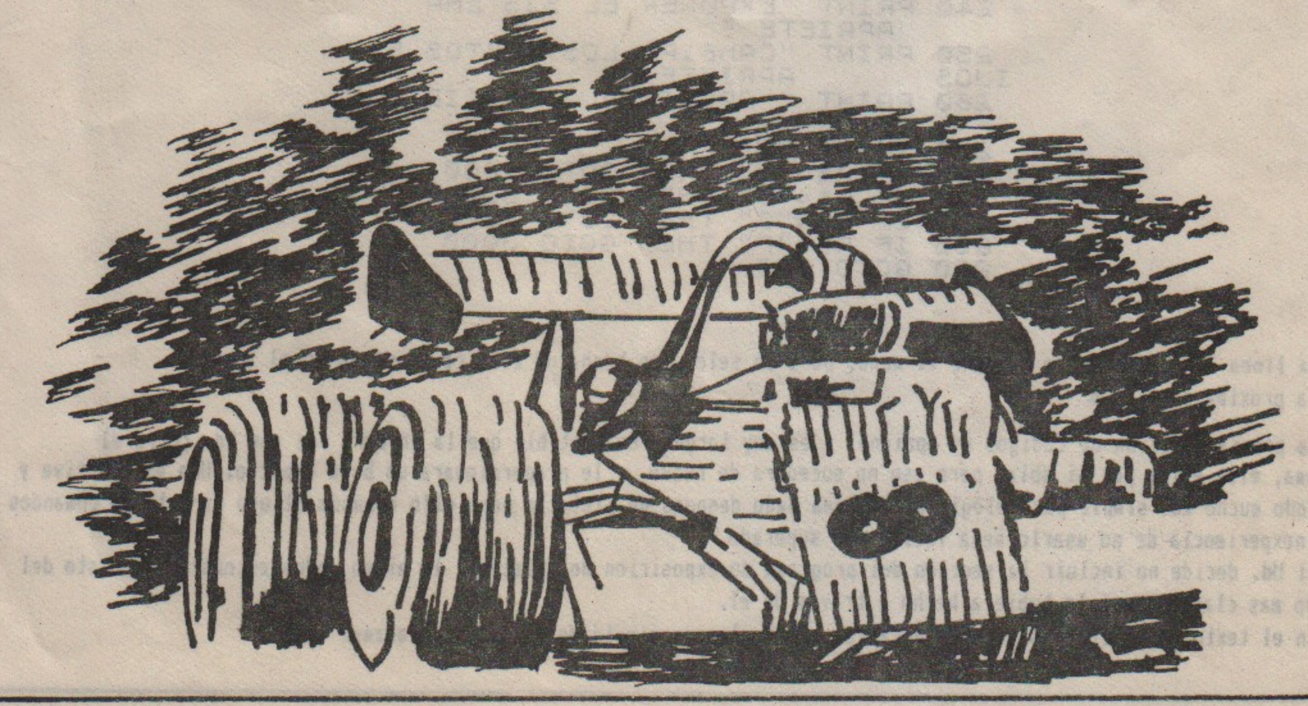
```
   1 GO SUB 1000: LET a=2: BORDE
R 1: PAPER 1: CLS
   3 PRINT AT 8,2; INK a;"
IRRERA CF"
   5 IF a=8 THEN BEEP .3,12: BEE
P .5,6: BEEP .3,15: GO TO 10
   6 LET a=a+1: FOR f=1 TO 4: LE
T o=.0123: BEEP o,2: BEEP o,4: B
EEP o,6: BEEP o,8: BEEP o,10: BE
EP o,12: BEEP o,14: BEEP o,16: B
EEP o,2: BEEP o,4: BEEP o,6: NEX
T f: GO TO 3
  10 INK 7: CLS : LET s=0: LET t
iempo=0
  15 PRINT AT 0,12;"CARRERA"''''"
(A)--IZQ.","(S)--DER."''''"(K)--A
RRIBA","(M)--ABAJO"
  17 PRINT '''"ES UNA CARRARA CON
TRA EL RELOJ"
  20 PRINT AT 18,2;"PRESIONE ALG
UNA TECLA PARA            EMPE
ZAR": PAUSE 0
  51 REM ***********************
  52 REM *  "A" en lineas      *
  53 REM * 210,240 y 260 de-   *
  54 REM * ben escribirse en   *
  55 REM *   MODO GRAFICO      *
  56 REM ***********************
  90 LET q=1: LET la=100: LET la
p=1: BORDER 1: PAPER 1: INK 9: C
LS
 100 FOR f=0 TO 31: PRINT AT 1,f
; PAPER 2;"*";AT 20,f;"*";: NEXT
f
 101 FOR f=1 TO 20: PRINT AT f,0
; PAPER 2;"*";AT f,31;"*";: NEXT
f
 103 FOR f=3 TO 19: PRINT AT f,3
; PAPER 6;"*";AT f,13;"*";AT f,1
9;"*";AT f,28;"*": NEXT f
 104 FOR f=2 TO 18: PRINT AT f,6
; PAPER 6;"*";AT f,16;"*";AT f,2
5;"*": NEXT f
 105 FOR f=7 TO 10: PRINT AT 6,f
; PAPER 3;"*";AT 12,f;"*";AT 18,
f;"*": NEXT f
 106 FOR f=9 TO 12: PRINT AT 3,f
; PAPER 3;"*";AT 9,f;"*";AT 15,f
;"*": NEXT f
 107 FOR f=20 TO 22: PRINT AT 3,
f; PAPER 4;"*";AT 9,f;"*";AT 15,
f;"*": NEXT f
 108 FOR f=22 TO 24: PRINT AT 6,
f; PAPER 4;"*";AT 12,f;"*";AT 18
,f;"*": NEXT f: GO TO 190
 120 FOR
 120 FOR f=3 TO 15 STEP 6: PRINT
 AT f,8; PAPER 3;"*";AT f,23; PA
PER 4;"*";: NEXT f
 121 FOR f=6 TO 18 STEP 6: PRINT
 AT f,11; PAPER 3;"*";AT f,21; P
APER 4;"*";: NEXT f: GO TO 200
 140 FOR f=4 TO 16 STEP 6: PRINT
 AT f,8; PAPER 3;"*";AT f,23; PA
PER 4;"*";: NEXT f
 141 FOR f=5 TO 17 STEP 6: PRINT
 AT f,11; PAPER 3;"*";AT f,21; P
APER 4;"*";: NEXT f: GO TO 200
 160 FOR f=3 TO 18 STEP 5: PRINT
 AT f,2; PAPER 6;"*";AT f,5;"*";
AT f,14;"*";AT f,17;"*";AT f,26;
"*";AT f,29;"*";: NEXT f
 161 FOR f=6 TO 18 STEP 6: PRINT
 AT f,1; PAPER 6;"*";AT f,4;"*";
AT f,15;"*";AT f,18;"*";AT f,27;
"*";AT f,30;"*";: NEXT f: GO TO
200
 180 FOR f=8 TO 14 STEP 6: PRINT
 AT f,8; PAPER 3;"*";AT f,23; PA
PER 4;"*";: NEXT f
 181 FOR f=7 TO 13 STEP 6: PRINT
 AT f,11; PAPER 3;"*";AT f,21; P
APER 4;"*";: NEXT f: GO TO 200
 194 POKE 23673,255: POKE 23672,
255
 196 PRINT AT 20,29; INK 1;"+";A
T 20,30;"+"
 198 PRINT AT 0,0;"MEJOR TIEMPO
";INT (tiempo);AT 21,0;"TIEMP.",
AT 21,13;"VUELTA"
 200 PRINT AT 21,20;lap: LET a=1
9: LET b=1
 205 LET ti=(256*PEEK 23673+PEEK
 23672)/50
 208 PRINT AT 21,6;INT ti
 210 PRINT AT a,b;"A";AT a,b;"A"
 AT a,b;"A";AT a,b;" ";
 220 LET b=b-(INKEY$="a")+(INKEY
$="s"): BEEP .001,3
 230 LET a=a-(INKEY$="k")+(INKEY
$="m")
 240 IF SCREEN$ (a,b)="*" THEN F
OR q=0 TO 7: PRINT AT a,b; FLASH
 1;"A": BORDER q: BEEP .09,RND*2
0: NEXT q: GO TO 700
 250 IF SCREEN$ (a,b)="+" THEN B
EEP .5,6: GO TO 500
 260 PRINT OVER 1;AT a,b;"A"
 300 GO TO 205
 500 LET la=la+20: LET lap=lap+1
 502 IF lap=6 THEN GO TO 600
 504 PRINT AT 21,20;lap
 506 GO TO la
 600 FOR f=0 TO 7: BEEP .002,RND
: BEEP .003,RND: BEEP .008,RND:
BEEP .006,RND: BORDER f: NEXT f:
 BORDER 7: PAPER 7: INK 0: CLS
```

```
 610>PRINT "UD. HA COMPLETADO LA
 CARRERA"'''"LA TERMINO EN ";INT
 (ti);"  SEGUNDOS"
 611 LET s=s+1
 612 IF s>1 THEN GO TO 630
 615 IF INT (ti)>tiempo THEN LET
 tiempo=INT (ti)
 620 PRINT '''"MEJOR TIEMPO DE C
ARRERA ";INT (tiempo)
 625 GO TO 730
 630 IF INT (ti)<tiempo THEN LET
 tiempo=INT (ti)
 640 GO TO 620
 700 BORDER 7: PAPER 7: INK 0: C
LS
 710 PRINT "UD. SE ESTRELLO CONT
RA"'"EL GUARD RAIL"'''"HA CORRID
O DURANTE ";INT (ti);"  SEG."
 720 PRINT '''"EN LA VUELTA ";s
p'''"HIZO EL MEJOR TIEMPO ";tiem
po
 730 PRINT '''"    OTRA CARRERA (
Y::N)"
 731 IF INKEY$="Y" OR INKEY$="y"
 THEN GO TO 50
 732 IF INKEY$="N" OR INKEY$="n"
 THEN GO TO 2900
 733 GO TO 731
1000 FOR a=0 TO 7: READ n: POKE
USR "A"+a,n: NEXT a: RETURN
1010 DATA 60,60,60,255,255,255,1
02,102
2900 BORDER 7: PAPER 7: CLS : LE
T q=1
3000 PRINT AT 10,12; INK q;"apla
usos"
3003 IF q=6 THEN LET q=1
3004 LET q=q+1: FOR f=1 TO 4: LE
T a=.012: BEEP a,-4: BEEP a,-6:
BEEP a,-8: BEEP a,-10: BEEP a,-1
2: BEEP a,-14: BEEP a,-16: BEEP
a,-2: BEEP a,-4: BEEP a,-6: NEXT
 f: GO TO 3000
5000 SAVE "carrera" LINE 0
```

Usted esta en la largada de las 500 millas de SINTAXNAPOLIS, donde por supuesto Ud. es el favorito. Tipee el juego y pronto estara manejando alrededor de la pista atravesando las chicanas tratando de llegar al final con el mejor tiempo. Si choca contra las barras se deslizara fatalmente. Los controles e instrucciones estan en el programa. Para calificarse en el campeonato debera tener por lo menos un SCORE de 32 segundos.

NOTA GRAFICA: el auto se dibuja con GRAPHICS "A".





CUENTAS PERSONALES

Este programa, el cual llevara todas las cuentas de su casa, le ahorrara tiempo, pero ese no es el beneficio mas importante que este paquete ofrece. La siguiente lista de trabajos puede ser codificada en el programa y cargarla le llevara solo unos segundos.

TRABAJOS QUE SERAN COMPUTADOS

1- Archivos con detalle de los pagos regulares y los datos de cuando caera su vencimiento.
2- Los cheques que seran deducidos proximamente.
3- Recordar los proximos pagos y ver que no se haya olvidado de agregar alguno.
4- Decidir cuando sus cheques por salario seran agregados, y costos fijos deducidos.
5- Permitir la posibilidad de pagos especiales dentro y fuera de las cuentas ya establecidas.
6- Hacer convenios para cambiar los costos fijos.

Para hacer siempre el trabajo, el programa deduce los costos fijos el mismo dia del mes en que acredita el salario.
Se ha usado como dia de pago el 25 de cada mes.
La maquina decide cuando preguntar por el monto del cheque de pago basandose en dos datos, buscados en la memoria.
Los datos entrados antes de dar el RUN al programa son recordados por la maquina y los nuevos datos se los preguntara ni bien Ud.haya cargado el programa. La primera parte del programa es un sistema para el manejo de caja. Puede ser cargado desde el cassette y podra usarlo como base de este programa.
La seccion siguiente al manejo de caja es el menu principal, el que aparecera en pantalla tan pronto como Ud. termine de cargar el programa.

```
 200 CLS
 210 PRINT "CUENTAS PERSONALES"
 220 PRINT "-------"
 230 PRINT "BALANCE HASTA LA FEC
HA       APRIETE U"
 240 PRINT "EXPONER EL SISTEMA
   APRIETE S"
 250 PRINT "CAMBIAR LOS COSTOS F
IJOS        APRIETE C"
 260 PRINT " ARCHIVAR    APRIETE
F"
 270 INPUT A$
 280 IF A$="U" THEN GOTO 2000
 282 IF A$="S" THEN GOTO 300
 284 IF A$="C" THEN GOTO 1200
 286 IF A$="F" THEN GOTO 3000
 290 GOTO 200
```

La linea 290 hace que se retorne al menu, pues la seleccion hecha no esta contenplada en el menu.
La proxima seccion d

La proxima seccion de codigos es opcional y es muy larga. Es factible que la primera vez que Ud. corra el programa, ella elija por si sola, pero eso no sucedera de nuevo, y le proporcionara un buen espacio. Una alternativa y un metodo mucho mas simple para elegir el sistema dado despues de listarlo pero ello demanda el uso directo de comandos y la inexperiencia de no usarlo sera facilmente superada.
Si Ud. decide no incluir la seccion del programa de exposicion del sistema, lo mismo podra encontrar el resto del listado mas claro que si lo hubiera hecho a traves de el.
En el texto son incluidos algunos detalles sobre almacenamiento de datos y cobranzas.

```
 300 CLS
 310 LET BL=0
 330 LET D$="MES VENCIDO"
 340 PRINT "ESTA EN CREDITO O PA
SADO DE GIRO CON EL BANCO? C/G"
 350 INPUT B$
 360 GOSUB 10
 370 IF B$="0" THEN GOSUB 30
 380 IF B$="C" THEN GOSUB 35
 390 IF B$<>"C" AND B$<>"0" THEN
 GOTO 300
```

Este programa permite llevar cuenta de hasta 10 costos fijos para ser pagados por mes, 10 que son pagados trimestralmente, y 10 que sean pagados una vez al año.

Los datos seran cargados en forma de vectores que seran definidos de la siguiente forma:

```
400 DIM Q(10)
410 DIM N(10)
420 DIM M(10)
```

Para cargar los elementos de los vectores se usan dos loops. Un loop se maneja con la letra "K", y el otro es controlado con la letra "J".

Las siguientes son las lineas en BASIC que hacen ese trabajo.

```
430 FOR K=1 TO 3
 440 FOR J=1 TO 10
 450 IF K=1 THEN PRINT "MENSUALM
ENTE";
 460 IF KA2 THEN PRINT "CUATRIME
STRALMENTE";
 470 IF K=3 THEN PRINT "ANUALMEN
TE";
 480 PRINT "*NO HAY ORDEN FIJA*"
J
 490 GOSUB 10
 500 IF AJ=0 THEN NEXT K
```

Esta seccion imprime las respuestas.

La maquina le preguntara por el monto de cada costo fijo, uno por vez. Si Ud. responde con cero, entonces asumira que todos los costos fijos del mes han sido entrados antes y pasara a pedirle los costos fijos trimestrales.

Si los 10 costos fijos trimestrales han sido entrados la maquina automaticamente pasara a pedirle los costos fijos anuales.

La maquina usa los valores cargados en "K" para decidir que es lo que hace.

```
510 IF K=2 THEN GOTO 590
520 IF K=3 THEN GOTO 690
530 IF K=4 THEN GOTO 800
540 GOSUB 70
570 NEXT J
580 NEXT K
```

Si "K" no es igual a dos o tres o aun a cuatro, entonces elegira uno de los montos de pago que fue cargado en el vector M elemento J. El arreglo esta cargado con informacion sobre pagos mensuales.

```
590 PRINT D$
620 INPUT A
630 PRINT TAB 8 ;"*MES*";A
650 IF A$<>"C" THEN GOTO 550
```

El proximo trabajo es cargar el monto de los pagos trimestrales y el mes durante el cual deberan ser pagados; ambos son almacenados en la misma variable.

Supongamos un monto regular de $a 999.99, entonces si a este monto lo dividimos por mil, el resultado sera menor que uno. Asi si Ud. debe pagar $a 103.75 el valor almacenado sera 0.10375.

Si entonces el numero del mes para el proximo pago es agregado al elemento, para el mes de Diciembre el valor que mantiene almacenado sera 12.1035. Para saber el mes se toma el valor entero del numero; para el monto es un poco mas complicado.

1) Toma el valor del entero y lo multiplica por 100000 y llama a esto A. 2) Multiplica el valor en el elemento por 100000 y lo llama B. 3)Resta: A-B y divide por 100.

Veamoslo:

1) 12*100000=1200000.......... ...A=1200000
2) 1210375*100000=1210375..... ...B=1210375
3) A-B=10375

Esto parece un poco largo pero en realidad ahorra espacio, y es un buen metodo para extraer informacion.

```
660 LET Q(J)=AJ/1000+A
670 NEXT J
680 NEXT K
```

Es decir si el mes y el monto de pago fueran almacenados en variables separadas tendriamos al menos 100 bytes extras para el almacenamiento.

Otra forma del programa que sigue a continuacion, guarda los valores en un vector (N) que mantiene las ordenes anuales.

```
690 PRINT D$
710 INPUT A
730 LET N(J)=AJ/1000+A
740 NEXT J
750 NEXT K
```

Al fin del loop controlado por "K" el contador debe ponerse en 4 y luego el loop no se inicia. La computadora reconoce esto como una senal en la linea 530 y salta a la linea 800 donde esta la rutina que muestra el estado de "costos fijos".

```
800 CLS
810 GOSUB 80
820 GOSUB 70
830 IF A$<>"C" THEN GOTO 400
840 CLS
```

Ir a 480, en la linea 630 tiene el efecto de redimensionar los vectores con la misma variable. Habiendo hablado ya de la linea 80 es hora que demos su listado.

```
80 PRINT "ORDENES FIJAS"
81 LET Z=100000
82 PRINT "NO." TAB 24;"£MES"
TAB 13;"£ TRIMESTRE" ; TAB 24
"£ ANUAL"
83 FOR J=1 TO 10
84 PRINT J; TAB 4; M(J) ;TAB 1
3 (INT (Q(J)*Z+.5)-INT Q(J)*Z)
100 ;TAB 20 ;INT Q(J) ; TAB 24
INT (N(J)*Z+.5)-INT N(J)*Z)/10
0 TAB 30; INT N(J
85 NEXT J
86 RETURN
```

La linea 84 es compleja pero hubiera sido peor de no ser por la linea 81 que evita el manejo de los "100000".

la explicacion ya hecha del almacenamiento y el metodo de extraccion sirve para entender esta linea.

La linea 900 inicia otro segmento que se basa en otras tres variables para mantener otro dato.

Este es el dato que debe recordarse desde la primera ejecucion (RUN) a la proxima y es el cual le ayudara a decidir cual pago se debe desde la ultima vez que fue usado.

```
900 GOSUB 60
910 LET D4=D1
930 LET D6=D3
940 CLS
```

Luego sigue la parte que carga en detalle de los libros de cheque y "libros de pagos".

```
1000 CLS
1010 PRINT "TECLEE EL NUMERO DE
SU CHEQUERA"
1020 INPUT C
1030 PRINT C
1040 PRINT "AORA EL SIGUIENTE NR
O. DE CHEQUE"
1050 INPUT C1
1070 PRINT "PROXIMAMENTE EL ULTI
MO NRO. DE CHEQUE EN LA CHEQUERA
"
1080 INPUT C2
1090 PRINT C2
1100 PRINT "Y EL NRO. DE PROXIMO
 PAGO ATRASADO"
1110 INPUT C3
1120 PRINT C3
1130 GOSUB 70
1140 IF A$<>"C" THEN GOTO 1000
1150 GOTO 200
```

Este es el fin de la rutina de "puesta a punto", una vez usada puede borrarse.

Como ya fue mensionado los "costos fijos" pueden no mantenerse fijos en el mismo nivel para siempre. Para ello hay una rutina que permite dichos arreglos. Las lineas desde 1200 hacen uso de la subrutina de la linea 80 para imprimir el mapa de "costos fijos".

```
1200 CLS
1210 PRINT "REVISADO"
1220 GOSUB 80
1230 PRINT
1240 PRINT "QUE ORDEN FIJA?"
1250 INPUT J
1260 PRINT J
1270 PRINT "MES(1) CUATRIMESTRE(
2) OR ANIO(3)?"
1280 INPUT A
1290 CLS
1300 PRINT "ENTREGAR VALOR CORRE
GIDO"
1310 GOSUB 10
```

Ahora la computadora tiene toda la informacion para seleccionar el correcto elemento y guardar el dato en el correspondiente vector. Las dos lineas siguientes seleccionan la apropiada rutina de carga.

```
1320 IF A=2 THEN GOTO 1380
1330 IF A=3 THEN GOTO 1430
```

Si A es otra cosa, esta debe ser para un costo fijo mensual.

```
1340 LET M(J)=AJ
1350 GOTO 200
1380 PRINT D$
1390 INPUT A
1400 LET Q(J)=AJ/1000+A
1410 GOTO 200
1430 PRINT D$
1440 INPUT A
1450 LET N(J)=AJ/1000+A
1460 GOTO 200
```

Ahora sigamos con un negocio regular.

```
2000 CLS
2010 PRINT "TRANSACCIONES DESDE"
;D4;"/";D5;"/";D6
2020 PRINT
2030 GOSUB 60
2040 IF (D3-D6)*365+(D2-D5)*30+D
1-D4<0 THEN GOTO 2030
```

Esta ultima linea prevee los cambios por atraso en tiempo. Todo tipo de dificultades pueden surgir. Los salarios pueden ser pagados una vez y contados dos. Si el interes debe ser calculado, entonces el nuevo dato no puede ser alcanzado por el loop que genera al interes ya que se estaria en un ciclo cerrado. La linea calcula el numero de dias que ha pasado y se niega a continuar si dicho numero es negativo.

La proxima seccion es un poco es compleja de escribir.

```
2050 PRINT D1;"/";D2;"/";D3
2060 IF D5=D2 AND D6=D3 THEN GOT
O 2180
2070 IF D4<25 THEN GOSUB 2130
2080 LET D4=1
2090 LET D5=D5+1
2100 IF D5>12 THEN LET D6=D6+1
2110 IF D5>12 THEN LET D5=D5-12
2120 GOTO 2060
2130 PRINT "TECLEE SALARIO POR M
ES*" D5;"/";D6
2140 GOSUB 10
2150 GOSUB 35
```

Habiendo decidido cual mes de pago a cargar, la computadora continua decidiendo cual de los "costos fijos" se deben.

Estamos en el medio de la rutina de busqueda que se inicia en la linea 2130. Esta rutina llama a otras tres, las cuales no han sido listadas todavia. La rutina de la linea 90 es la que decide sobre "costos fijos".

```
2160 GOSUB 90
2170 RETURN
2180 IF D4>24 OR D1<25 THEN GOTO
 2200
2190 GOSUB 2130
  90 FOR J=1 TO 10
  91 LET AJ=M(J)
  92 GOSUB 30
  93 NEXT J
  95 FOR J=1 TO 10
  96 GOTO 120
  97 LET AJ=(Q(J)-INT Q(J))*1000
  98 GOSUB 30
  99 NEXT J
 100 FOR J=1 TO 10
 101 IF INT N(J)<>D5 THEN GOTO 1
04
 102 LET AJ=(N(J)-INT N(J))*1000
 103 GOSUB 30
 104 NEXT J
 105 RETURN
```

Ud. debe recordar que unicamente un mes del trimestre en el que debe ser pagado el "costo fijo" puede ser cargado el arreglo.

La seccion del programa que comienza en la linea 120 usa ese mes para chequear el numero de mes si alguno de los otros tres no han aparecido. Si el presente numero de mes, cargado en la maquina, es alguno de los cuatro meses en el que debe ser hecho el pago, entonces el programa va a la linea 97 y deduce el pago para el balance. En cualquier otro mes, esta seccion es salteada y el control es enviado a la linea 99.

Los pagos anuales son faciles. Ellos deben ser pagados solo cuando el"INT N(J)" (la parte entera del valor del elemento J del vector N), es igual al numero de mes presente.

```
 120 LET A=INT Q(J)
 130 IF D5=A OR D5=A+3 OR D5=A+6
OR D5=A+9 OR D5=A-3 OR D5=A-6 O
R D5=A-9 THEN GOTO 97
 140 LET A=0
 150 GOTO 99
```

El resto del listado es muy simple, comparado con el resto.

Primero, todos los cheques que han sido escritos desde la ultima vez que el balance fue hecho, seran deducidos. Si un valor 0 es entrado, la computadora preguntara si el cheque fue cancelado; si la respuesta es "NO" entonces ella tomara el "INPUT" como senal de que el ultimo cheque ha sido destruido, y preguntara por los pagos con atraso.

```
2200 CLS
2210 GOSUB 45
2220 PRINT "NRO. DE CHEQUERA*";C
2230 PRINT "TECLEE EL VALOR DEL
NRO. DE CHEQUE*";C1
2240 GOSUB 10
2250 GOSUB 30
```

```
2260 IF AJ=0 THEN GOTO 2450
2270 IF C1=C2 THEN PRINT "SI UD.
 TIENE DETALLES DE LA CHEQUERA"
2280 IF C1=C2 THEN PRINT C+1;"#L
UEGO TECLEE D,DE OTRA FORMA TECL
EE ENTER"
2290 IF C1=C2 THEN INPUT A$
2300 IF C1=C2 AND A$="D" THEN GO
TO 2340
2310 IF C1=C2 THEN GOTO 2475
```

Todo esto necesita una o dos palabras de explicacion. Si C1 es igual a C2 entonces el proximo cheque es el ultimo en la chequera. Si la proxima chequera ya le llego del banco, entonces los datos podran ser cargados despues de pulsar la "D". Si, de otra forma, el banco se demora en enviarle la nueva chequera, Ud. no podra escribir nada y el programa seguira con su flujo.

```
2320 LET C1=C1+1
2330 GOTO 2200
2340 LET C=C+1
2350 CLS
2360 PRINT "TECLEE EL PRIMER NRO
. DE CHEQUE EN LA CHEQUERA #";C
2370 INPUT C1
2380 PRINT C1
2390 PRINT "AHORA EL ULTIMO"
2400 INPUT C2
2410 PRINT C2
2420 GOSUB 70
2430 IF A$<>"C" THEN GOTO 2350
2440 GOTO 2200
```

Ahora viene la rutina de cancelacion de cheques:

```
2450 PRINT "SI ESE FUE UN CHEQUE
 CANCELADO TECLEE C,DE OTRA FORM
A TECLEE ENTER"
2460 INPUT A$
2470 IF A$="C" THEN LET C1=C1+1
2475 CLS
2480 IF A$="C" THEN GOTO 2200
```

Ahora que todas las posibilidades ya han sido ingresadas, comenzamos el balance con la linea 2490.

```
2490 GOSUB 45
2500 PRINT "PAGOS EN CTA. CORRIE
NTE"
2510 PRINT
2520 PRINT "TECLEE VALOR DEL NRO
.DE PAGO ATRASADO#",C3
2530 GOSUB 10
2540 IF AJ=0 THEN GOTO 2600
2550 GOSUB 35
2560 LET C3=C3+1
2570 GOTO 2500
```

Y ahora los pagos especiales que le hagan o que Ud. haya hecho sobre montos que no han sido trabajados con cheques.

```
2600 CLS
2610 GOSUB 45
2620 PRINT "ALGUN DESEMBOLSO ESP
ECIAL? S/N "
2630 INPUT A$
2640 IF A$<>"Y" THEN GOTO 2690
2650 GOSUB 10
2660 GOSUB 30
2670 GOTO 2600
2690 CLS
2700 PRINT "ALGUN DEPOSITO ESPEC
IAL? S/N"
2710 INPUT A$
2720 IF A$<>"Y" THEN GOTO 200
2730 GOSUB 10
2740 GOSUB 35
2750 GOTO 2690
```

Finalmente, la rutina de comienzo automatico sera impresa despues del balance general.

```
3000 CLS
3010 PRINT TAB 8;"BALANCE ESPECI
AL"
3020 PRINT
3030 GOSUB 45
3040 PRINT AT 10,10;"PREPARE EL
GRABADOR"
3045 PRINT TAB 10;"TECLEE ENTER"
3050 INPUT A$
3060 LET D4=D1
3070 LET D5=D2
3080 LET D6=D3
3090 CLS
3100 SAVE "CAJA"
3110 GOTO 200
```

Si, en cambio, Ud. ha omitido la rutina de inicio, hay tro trabajo por hacer antes de que el programa sea corrido sin que aborte.

PROCEDIMIENTO PARA EL COMIENZO

Antes que todo, deletee las lineas 240 a la 282 de la seccion del menu y entonces teclee estos comandos sin numero de linea. La computadora ejecutara en forma correcta y la informacion sera recordada desde un "RUN" a otro.

Mire todo el listado para estar seguro ue toda la informacion ha sido entrada antes de comenzar.

1) LET BL=(balance de su banco)/ENTER.
2) DIM M(10)/ENTER.
3) DIM N(10).
4) DIM Q(19)/ENTER.
5) LET D4=(numero de dia)/ENTER.
6) LET D5=(numero de mes)/ENTER.
7) LET D6=(los dos ultimos digitos del ano)/ENTER.

Teclee GO TO 200 y el menu principal aparecera. Presione "C" para cambiar los costos fijos y luego cambie por cero y mes cero los montos pagables y el mes en que vence el pago.

Los pagos trimestrales necesitan solo un numero de mes, no cuatro.

Su programa esta listo para correr.

VARIABLES USADAS

Para el manejo de caja:

- AJ ajuste de caja
- A$ para seleccionar
- BL balance
- D1 numero de dia
- D2 numero de mes
- D3 numero de año (solo los dos ultimos digitos)

Variables especificas:

- B$ variable de seleccion para ser usada cuando A$ permanezca fijo
- A cualquier input que no sea permanente
- D$ mes de vencimiento
- C numero de chequera
- C1 numero del proximo cheque
- C2 ultimo cheque en la chequera
- C3 numero del proximo pago atrasado
- D4 a D6 para cargar datos secundarios
- J y K contadores
- Z ajusta matematicamente a 100000
- Q, N y M arreglos dimensionados en 10 para cargar los costos fijos

15 Pulsos

```
  3 DIM B(8)
  5 DIM A(16)
 10 GOSUB 300
 15 LET M=0
 20 FOR I=1 TO 16
 30 LET A(I)=0
 40 NEXT I
 50 FOR I=1 TO 16
 60 LET R=RND*16
 70 IF NOT A(R)=0 THEN GOTO 60
 80 LET A(R)=I
 90 NEXT I
100 GOSUB 500
110 IF F=1 THEN GOTO 20
120 GOSUB 600
130 PRINT
140 PRINT "SU JUGADA"
150 INPUT X
152 LET C=X
155 IF X=0 THEN STOP
160 GOSUB 400
170 GOSUB 700
180 IF NOT F=0 THEN GOTO 210
185 PRINT
190 PRINT C;"JUGADA ILEGAL,VUEL
VA A JUGAR"
200 GOTO 150
210 LET A(X+F)=A(X)
220 LET A(X)=16
230 GOTO 800
240 LET M=M+1
250 GOTO 120
300 LET B(1)=2
310 LET B(2)=4
320 LET B(3)=5
330 LET B(4)=7
340 LET B(5)=10
350 LET B(6)=12
360 LET B(7)=13
370 LET B(8)=15
380 RETURN
390 REM CONVERTIR EL NRO EN UN
LUGAR DEL ARREGLO
400 FOR I=1 TO 16
410 IF A(I)=X THEN GOTO 430
420 NEXT I
430 LET X=I
440 RETURN
450 REM VERIFICA SI ES UNA SOLU
CION POSIBLE
500 LET F=1
510 LET S=0
520 FOR I=1 TO 15
530 FOR J=I+1 TO 16
540 IF A(I)>A(J) THEN LET S=S+1
550 NEXT J
560 NEXT I
565 FOR I=1 TO 8
570 IF A(B(I))=16 THEN LET S=S+
1
575 NEXT I
585 IF (S/2)*2=S THEN LET F=0
590 RETURN
595 REM MOSTRAR TABLA DE JUEGOS
600 CLS
610 PRINT ,"QUINCE PULSOS"
615 PRINT
620 LET I=1
625 PRINT
630 FOR Y=1 TO 4
632 IF A(I)<10 THEN PRINT " ";
635 IF A(I)=16 THEN PRINT "  ";
640 IF NOT A(I)=16 THEN PRINT A
(I);
645 PRINT " ";
650 LET I=I+1
655 NEXT Y
660 PRINT
662 PRINT
665 IF I=17 THEN RETURN
670 GOTO 625
680 REM PROBAR PARA JUGADA LEGA

700 LET F=0
710 IF X+1>16 THEN GOTO 725
720 IF A(X+1)=16 THEN LET F=1
725 IF X-1<0 OR X-1=0 THEN GOTO
735
730 IF A(X-1)=16 THEN LET F=-1
735 IF X+4=16 THEN GOTO 745
740 IF A(X+4)>16 THEN LET F=4
745 IF X-4<0 OR X-4=0 THEN GOTO
760
750 IF A(X-4)=16 THEN LET F=-4
760 RETURN
770 REM CHEQUEANDO EL GANADOR
800 FOR I=1 TO 16
810 IF NOT A(I)=I THEN GOTO 240
820 NEXT I
830 GOSUB 600
840 PRINT
850 PRINT
860 PRINT "LE LLEVO SOLO";M;"MO
VIDAS"
870 STOP
```

Este programa es de un juego que fue inventado por Sam Loyd en 1878 y que consiste de una matriz de 4*4 que tiene 15 bloques sin numerar. El objeto del juego es terminar con todos los bloques numerados. Suena facil pero no lo es.

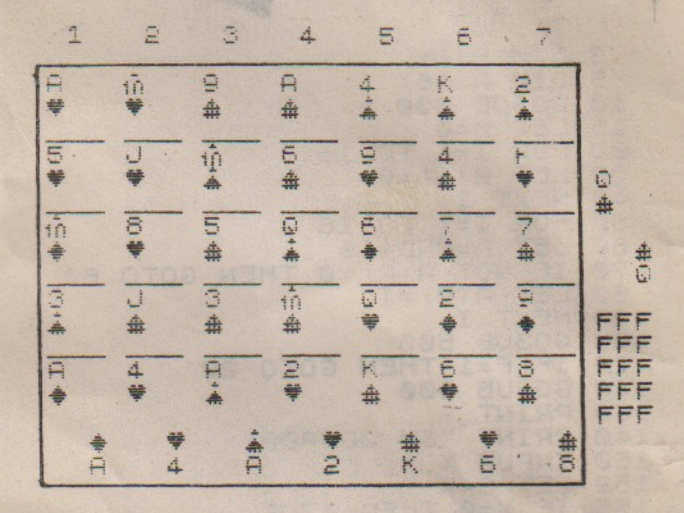
No todas las combinaciones de comienzo factible son: 20, 922, 789, 888, y 000, asi que el juego las chequea. Si no es su jugada de comienzo la computadora blanquea el display antes que Ud. pulse. Es por ello que antes de mostrar cualquier cosa en pantalla se demora algunos segundos. El programa tambien chequea por movidas ilegales en cada entrada.

Con cada entrada, la computadora chequea para ver si gano. Si Ud. lo hace ella lo felicitara y le dira cuantas jugadas le tomo hacerlo.

Si logra frustrarlo, pulse 0 (cero) y el juego termina.

Solitario

```
  10 REM "SOLITARIO"
  20 DATA 0,54,127,127,62,28,8,0
  30 DATA 0,8,28,62,127,62,28,8
  40 DATA 0,8,28,42,127,42,127,4
2,8,28
  50 DATA 0,8,28,62,127,42,8,28
  60 DATA 0,70,201,73,73,73,230,
0
  70 DATA 153,90,60,255,255,60,9
0,150
  80 FOR i=144 TO 149: FOR j=0 T
O 7
  90 READ a: POKE USR CHR$ i+j,a
 100 NEXT j: NEXT i
 110 BORDER 4: PAPER 4: INK 0: C
LS
 120 DIM a(6,7): DIM b(17): LET
b(1)=17: LET b$=""
 130 LET a$="": FOR i=1 TO 52: L
ET a$=a$+CHR$ i: NEXT i
 140 PRINT AT 11,11;"BARAJANDO"
 150 FOR I=1 TO 52
 160 LET w=INT (RND*52+1)
 170 IF CODE a$(w)<>0 THEN GO TO
 200
 180 LET w=w+1: IF w=53 THEN LET
 w=1
 190 GO TO 170
 200 LET b$=b'$+CHR$ w
 210 LET a$(w)=CHR$ 0
 220 NEXT i
 230 LET a=0
 240 FOR i=1 TO 7
 250 FOR j=2 TO 6
 260 LET a=a+1
 270 LET a(j,i)=CODE b$(a)
 280 NEXT j: NEXT i
 290 FOR i=1 TO 7: LET a(1,i)=6:
 NEXT i
 300 LET a=a+1: LET b=CODE b$(a)
 310 FOR i=2 TO 17: LET a=a+1:'L
ET b(i)=CODE b$(a): NEXT i
 320 CLS : PRINT "  ";: FOR i=1
TO 7: PRINT STR$ i;"  ";: NEXT
 : PRINT
 330 FOR i=2 TO 5: PRINT
 340 FOR j=1 TO 7  LET a=a(i,j):
 GO SUB 800
 350 PRINT " "; PAPER 7;b$;"  ";
 360 NEXT j
 370 PRINT
 380 FOR j=1 TO 7: LET a=a(i,j):
 GO SUB 800
 390 PRINT " "; PAPER 7;a$;"  ";
 400 NEXT j
 410 PRINT : FOR j=1 TO 7: PRINT
 " "; PAPER 7;"___";: NEXT j
 420 NEXT i
 430 PRINT : FOR j=1 TO 7: LET a
=a(6,j): GO SUB 800: PRINT " ";
PAPER 7;b$;"  ";: NEXT j
 440 PRINT : FOR j=1 TO 7: LET a
=a(6,j): GO SUB 800: PRINT " ";
PAPER 7;a$;"  ";: NEXT j
```

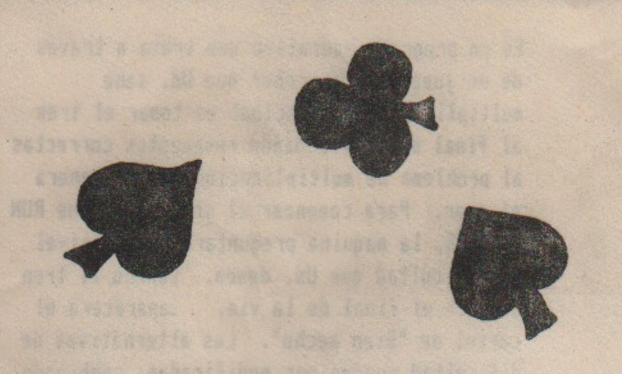


Cartas en el monton: 16

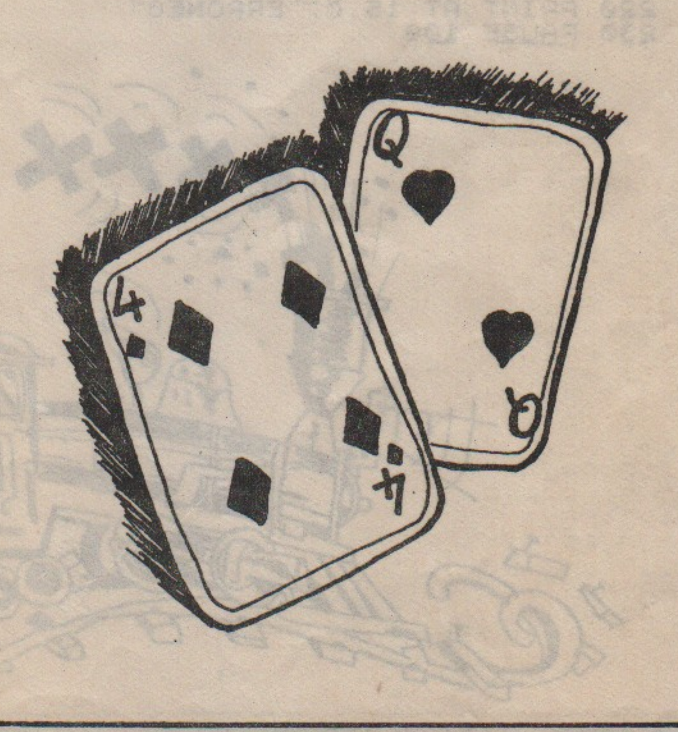
La idea es sacar todas las cartas de la pila del costado. Una carta se puede sacar si es la anterior o la posterior a la carta que esta en la pila. Ejemplo: si en la pila hay un 4 se puede sacar de las columnas un 3 o un 5. De cada columna solo se puede sacar la ultima carta. Para hacerlo presione la tecla correspondiente al numero de columna de la que quiere sacar. Si no tiene ninguna posibilidad de sacar cartas debera pedir una del monton presionando el 0 (cero). El juego termina cuando logro dejar vacias todas las columnas de cartas o bien porque ya no le quedan mas cartas en el monton (tiene 16 cartas para todo el juego). Buena suerte!

NOTA GRAFICA: los graficos se obtienen de:

GRAPHICS "A" corazon
GRAPHICS "B" diamante
GRAPHICS "C" trebol
GRAPHICS "D" pica



```
 450>PRINT : FOR j=1 TO 7: PRINT
" "; PAPER 7;"  ";: NEXT j
 460 PRINT : FOR j=1 TO 7: LET a
=a(5,j): GO SUB 800: PRINT " ";
PAPER 7;" ";a$;: NEXT j
 470 PRINT : FOR j=1 TO 7: LET a
=a(6,j): GO SUB 800: PRINT " ";
PAPER 7;" ";b$;: NEXT j
 480 PRINT AT 21,1; INK 0;"Carta
s en el monton: 15"; PAPER 7
 490 GO SUB 860
 500 GO SUB 880: GO SUB 890
 510 LET a$=INKEY$
 520 IF a$<"0" OR a$>"7" THEN GO
 TO 510
 530 LET c=VAL a$: IF NOT c THEN
 GO TO 730
 540 IF a(1,c)=1 THEN BEEP 1,0:
GO TO 510
 550 LET a=a(a(1,c),c): GO SUB 8
00: LET d=w
 560 LET a=b: GO SUB 800
 570 IF d=13 AND w=1 THEN GO TO
600
 580 IF d=1 AND w=13 THEN GO TO
600
 590 IF ABS (d-w)<>1 THEN BEEP 1
,0: GO TO 510
 600 BEEP .25,30
 610 LET b=a(a(1,c),c): GO SUB 8
60
 620 LET a(1,c)=a(1,c)-1: LET d=
(a(1,c)-1)*3+1
 630 LET a=a(a(1,c),c): GO SUB 8
00: LET c=(c-1)*4+1
 640 IF d=1 THEN FOR i=2 TO 6: P
RINT AT i,c; PAPER 4;"   ": NEXT
 i: GO TO 670
 650 PRINT AT d,c;"   ";AT d+1,c
;" ";a$;AT d+2,c;"  ";b$
 660 PAPER 4: PRINT AT d+3,c;"
";AT d+4,c;"   ";AT d+5,c;"   ":
 PAPER 7
 670 FOR i=1 TO 7: IF a(1,i)<>1
THEN GO TO 510
 680 NEXT i: PAPER 4: INK 0: PRI
NT AT 11,3;"FELICIDADES, LO CONS
IGUIO!!"
 682 FOR d=1 TO 20
 684 SOUND 6,15;7,7;8,16;9,16;10
,16;12,16;13,0
 686 PAUSE 6
 688 NEXT d
 690 PAPER 4: INK 0: PRINT AT 13
,3;"Desea intentarlo de nuevo?"
 691 SOUND 7,56
 692 SOUND 0,68;1,3;8,12
 693 PAUSE 60
 694 SOUND 2,151;3,2;9,12
 695 PAUSE 60
 696 SOUND 4,46;5,2;10,12
 697 PAUSE 100
 698 SOUND 0,0;1,0;2,0;3,0;4,0;5
,0
 700 LET a$=INKEY$: IF a$="s" OR
a$="S" THEN GO TO 110
 710>IF a$<>"n" AND a$<>"N" THEN
 GO TO 700
 720 STOP
 730 BEEP .25,30: IF b(1)=1 THEN
 GO TO 770
 740 LET b=b(b(1)): GO SUB 860:
LET b(1)=b(1)-1
 750 GO SUB 880: IF b(1)>1 THEN
GO SUB 890
 760 PRINT AT 21,22; PAPER 7; IN
K 0;b(1)-1;" ": GO TO 510
 770 LET c=0: FOR i=1 TO 7: LET
c=c+a(1,i): NEXT i: LET c=c-7
 780 PAPER 4: INK 0: PRINT AT 11
,5;"Perdio por ";c;" cartas"
 782 SOUND 7,62;8,15
 783 FOR I=50 TO 100
 784 SOUND 0,I: PAUSE 3
 785 NEXT I
 786 SOUND 6,6;7,7;8,16;9,16;10,
16;12,56;13,8
 787 PAUSE 90
 788 SOUND 8,0;9,0;10,0
 790 GO TO 690
 800 LET w=INT ((a-1)/13)
 810 LET a$=("♥" AND NOT w)+("♦"
AND w=1)+("♣" AND w=2)+("♠" AND
 w=3)
 820 INK 2: IF w>1 THEN INK 0
 830 LET w=a-w*13
 840 IF w<10 AND w<>1 THEN LET b
$=STR$ w: RETURN
 850 LET b$=("10" AND w=10)+("J"
AND w=11)+("Q" AND w=12)+("K" AN
D w=13)+("A" AND w=1): RETURN
 860 LET a=b: GO SUB 800: PRINT
AT 5,29; PAPER 4;"   ": PRINT AT
 6,29;b$;"  ";AT 7,29;a$;"  ";
 870 PRINT AT 8,29;"   ";AT 9,29
;"  ";a$;AT 10,29;"  ";b$: RETUR
N
 880 PAPER 4: FOR i=12 TO 16: PR
INT AT i,29;"   ": NEXT i: PAPER
 7: RETURN
 890 PAPER 2: INK 7: FOR i=12 TO
 16: PRINT AT i,29;"FFF": NEXT i
: PAPER 7: RETURN
1000 FOR n=0 TO 7
1100 INPUT fila: POKE USR "E"+n,
fila
1200 NEXT n
```



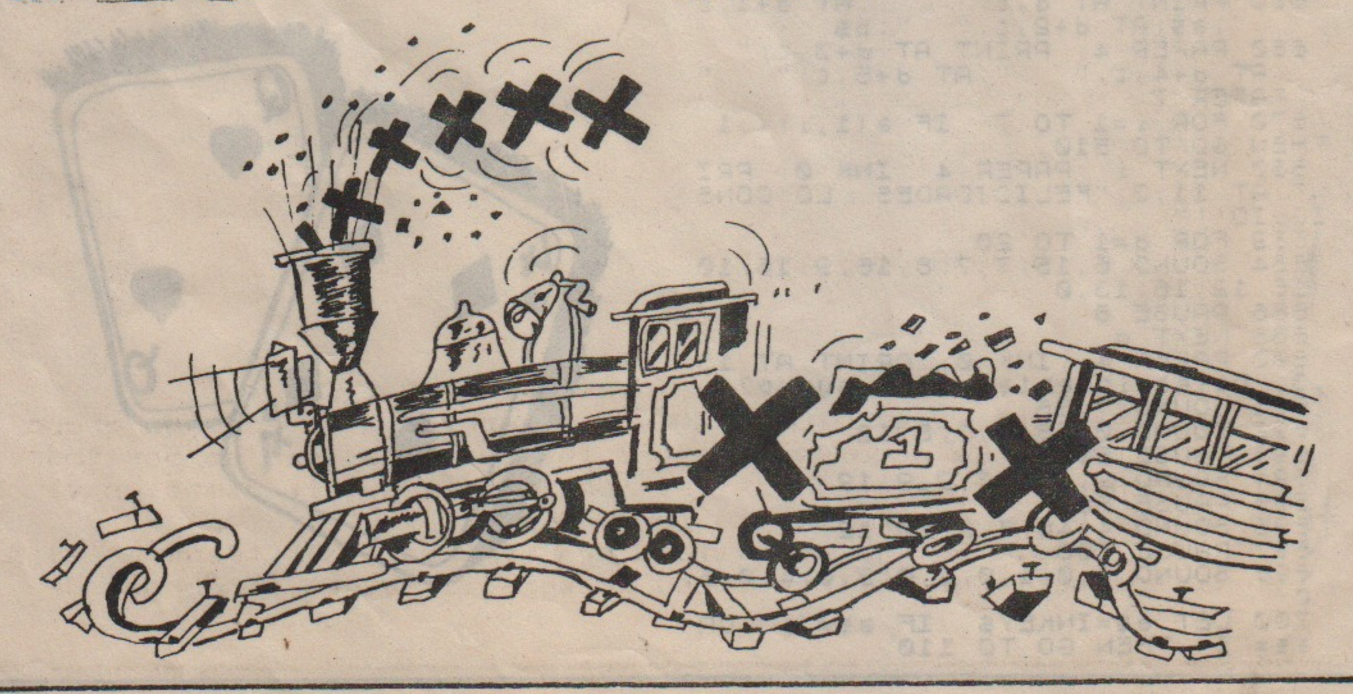
TREN DE MULTIPLICACION

Es un programa educativo que trata a traves de un juego de comprobar que Ud. sabe multiplicar. Lo principal es tomar el tren al final de la via dando respuestas correctas al problema de multiplicacion que se genera al azar. Para comenzar el juego presione RUN y ENTER, la maquina preguntara por el nivel de dificultad que Ud. desea. Cuando el tren alcance el final de la via, aparecera el cartel de "Bien hecho". Las alternativas de dificultad pueden ser modificadas, cambiando de la linea 50 a la 80.

NOTA GRAFICA:

Linea 100- Shift R- 3 espacios- Shift B- 1 espacio.
Linea 110- Shift B- 3 espacios- Shift 5.
Linea 120- Shift R- Shift Eespacio- Shift R- Shift E.
Linea 130- 14 guiones.

```
 10 LET Y=10
 20 LET X=0
 30 PRINT AT 3,0;"MANIA 1  OR 2"
 40 INPUT A$
 50 LET A=INT (RND*20)
 60 LET B=INT (RND*20)
 70 LET C=INT (RND*1)
 80 LET D=INT (RND*10)
 90 CLS
100 PRINT AT Y,X;"▀ ▄ █"
110 PRINT AT Y+1,X;" ███ "
120 PRINT AT Y+2,X;" ▀▀▀ "
130 PRINT AT 13,0;"--------------"
140 PRINT AT 0,0;"TOMAR AL FINAL"
150 IF A$="1" THEN PRINT AT 5,0
A;"*";B
160 IF A$="2" THEN PRINT AT 5,0
C;"*";D
170 LET H=A*B
180 LET K=C*D
190 INPUT Z
200 IF A$="1" AND Z=H THEN GOTO
250
210 IF A$="2" AND Z=K THEN GOTO
250
220 PRINT AT 15,0;"ERRONEO"
230 PAUSE 100
240 GOTO 50
250 PRINT AT 9,X+2;"▒▄"
260 LET X=X+1
270 IF X<10 THEN GOTO 300
275 PRINT AT 12,15;"█"
280 PRINT AT 13,15;"█.BIEN HECHO"
290 STOP
300 PAUSE 150
310 GOTO 50
```



ENUNCIADO GO TO

El siguiente es un programa que calcula el producto y el cociente de dos numeros, e imprime el resultado.

```
10 READ B,C
20 LET A1=B*C
30 PRINT "EL PRODUCTO ES"
         A1
40 LET A2=B/C
50 PRINT "EL COCIENTE ES"
         A2
60 DATA 36,9
70 END
```

Tal como esta el programa, cada vez que queramos repetir el calculo para un nuevo par de numeros, tendremos que volver a correr el programa con un nuevo enunciado DATA. Seria conveniente que pudieramos hacer que la computadora repitiera el calculo con otras cantidades, sin nuestra intervencion. Si insertamos la linea

```
55 GO TO 10
```

podemos hacer que el flujo vuelva a la linea 10 cuando llegue a la 55, y vuelva a empezar la secuencia del programa con

```
10 READ B,C
```

Si luego modificamos el enunciado DATA:

```
36 DATA 36,9,20,5
```

en la segunda corrida la maquina leera para B y C, respectivamente, los valores 20 y 5 e imprimira el segundo grupo de resultados. La maquina repetira los lazos de retorno por todo el programa hasta agotar la lista de numeros que le dimos en el enunciado DATA, y el programa terminara.

Asi pues, "GO TO" equivale a una bifurcacion o transferencia en la secuencia de instrucciones (cambia la secuencia normal de ejecucion).

ENUNCIADO FOR

El ciclo FOR es una orden para repetir un conjunto de enunciados para cada valor del contador, en el rango especificado. Estos ciclos se controlan contandolos, pues se sabe de antemano la cantidad de repeticiones necesarias.

```
10 FOR I=1 TO 10
20 PRINT I
30 NEXT I
40 END
```

"I" se denomina contador de ciclo y puede ser cualquier variable sin subindice.

Este enunciado FOR ordena la ejecucion de todos los enunciados siguientes hasta el enunciado NEXT, para I=1, I=2, etc., hasta I=10. El enunciado NEXT I abarca todos los que seran ejecutados y repetidos para cada valor de I. Siempre tiene que haber un enunciado NEXT correspondiendo a un enunciado FOR.

Despues del signo igual en un enunciado FOR hay que especificar un valor inicial del contador de ciclo. Despues de la palabra TO, hay que dar el valor final del contador de ciclo. Tanto el valor inicial como el final, pueden especificarse como una constante con signo, o sin el, o como una expresion.

Se supone que el incremento del contador de ciclo es de uno. Si se desea un incremento distinto de uno, debera especificarse a traves de un STEP. Estan permitidos STEP negativos (cuando queremos decrementar el contador).

```
10 FOR I=1 TO 10 STEP 2
20 PRINT I
30 NEXT I
40 END
```

```
10 FOR I=1 TO N/2 STEP -1
20 PRINT I
30 NEXT I
40 END
```

Si el valor inicial es mayor que el final y el incremento es positivo, o si es menor que el final pero el incremento es negativo, no se ejecutara el ciclo.

El siguiente enunciado que se ejecutara, en tales casos, sera el que siga al enunciado NEXT correspondiente.

Quiza Ud. este pensando en incluir un ciclo FOR-NEXT dentro otro; es decir, anidar ciclos (ponerlos unos dentro de otros). Esto es posible solo si se lo hace en forma correcta: no deben enlazarse ni superponerse en forma parcial y nunca debe quedar solo un enunciado FOR, dentro de un ciclo exterior mas grande, sin su NEXT correspondiente. El enunciado FOR siempre debe estar

acompañado por su correspondiente enunciado NEXT, dentro del ciclo mayor.

```
10 FOR X=1 TO 4
20 PRINT X
30 FOR Y=1 TO 3
40 PRINT Y
50 NEXT Y
60 NEXT X
70 END
```

ENUNCIADO IF

Las decisiones en BASIC, son tomadas a traves del enunciado IF.

Este consta de tres partes: la palabra IF (el si condicional) que identifica el enunciado, una relacion logica que puede ser verdadera o falsa, y la palabra THEN (entonces) seguida de la accion a tomar cuando la relacion tenga el valor logico "verdadero".

Las relaciones que pueden usarse son:

- = igual a
- <> no es igual a
- < menor que
- <= menor o igual a
- > mayor que
- >= mayor o igual a

Algunos ejemplos:

```
40 IF B-4*A>A-C/2 THEN GO ...     TO 200

30 IF D=3 THEN LET C=A+B

10 IF T<>4 THEN LET A=B*T

80 IF A*C<D THEN GO TO 50
```

Las dos aplicaciones mas usuales del enunciado condicional IF son: 1) crear una bifurcacion hacia otra parte del programa como resultado de una prueba (instrucciones 40 y 80).

2) incluir o excluir un enunciado intermedio, basandose en el resultado de una prueba (instrucciones 30 y 10).

EJERCICIOS

1) Hallar el mayor de un conjunto de 20 numeros e imprimirlo.

2) Realice la entrada y comprobacion de eco de dos valores, b y c, y halle la raiz de la ecuacion bx+c=0.

i) Si b=0 y c<>0, imprima la salida "bx+c=0 no tiene raiz".

ii) Si b=0 y c=0, imprima la salida "todo numero real satisface la ecuacion".

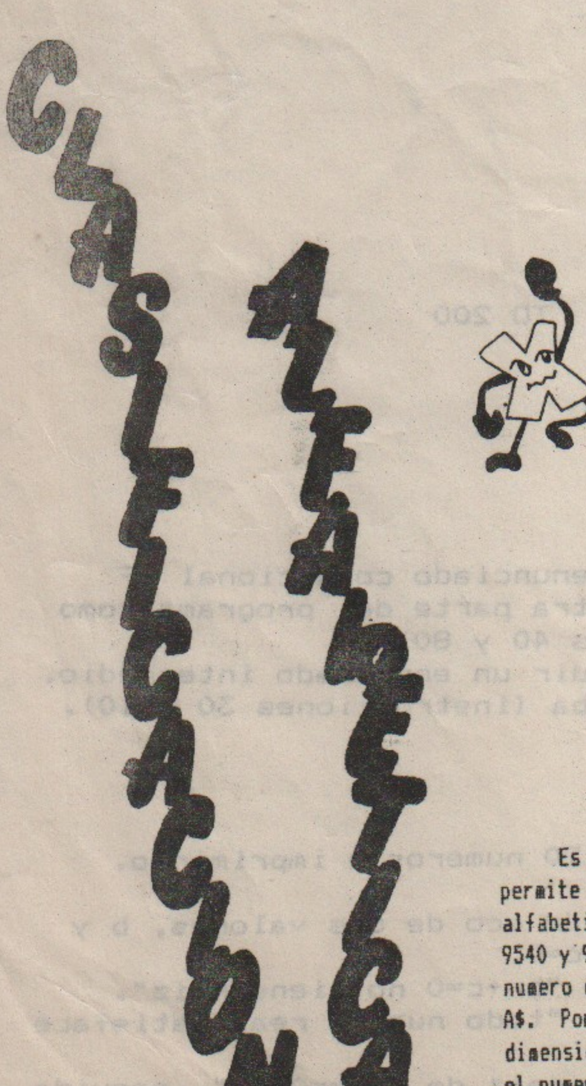
iii) Si b<>0, imprima la salida "la raiz de bx+c=0 es", seguida de la raiz.

SOLUCION DE LOS EJERCICIOS DE LA EDICION ANTERIOR

```
1) 10 READ R
   20 DATA 2
   30 PRINT AT1,1;"RADIO",
            AT1,11;"PERIMETRO"
            AT1,22;"AREA"
   40 LET A=2*Pi*(R↑2)
   50 LET CIR=2*Pi*R
   60 PRINT AT3,1;R,
            AT3,11;CIR,
            AT3,22;A
   70 END
```

```
2) 10 INPUT A,B
   20 LET X=A*B
   30 LET Y=A/B
   40 PRINT AT8,10;"X=";X
   50 PRINT AT8,20;"Y=";Y
   60 END
```

TK-83/ TK-85 TS 1000/1500




```
9500 FAST
9530 LET A=1
9540 LET N=100
9550 IF 2**A>N THEN GOTO 9580
9560 LET A=A+SGN A
9570 GOTO 9550
9580 LET F=2**A-1
9590 LET F=INT (F/2)
9600 IF NOT F THEN GOTO 9750
9610 LET D=N-F
9620 LET B=SGN A
9630 LET A=B
9635 LET E=A+F
9650 IF A$(A)>A$(E) THEN GOTO 97
00
9660 LET B=B+SGN B
9670 GOTO (B>D AND 9590)+(9620 A
ND B<=D)
9700 LET B$=A$(A)
9710 LET A$(A)=A$(E)
9720 LET A$(E)=B$
9730 LET A=A-F
9740 GOTO (9660 AND A(1)+(9630 A
ND A>=1))
9750 SLOW
9760 CLS
9770 FOR F=1 TO 100
9780 SCROLL
9790 PRINT A$(F)
9800 NEXT F
9810 RETURN
```

Es una rutina que nos permite ordenar A$ en forma alfabetica. Altere las lineas 9540 y 9770 de acuerdo con el numero de elementos que tenga A$. Por ejemplo si Ud. dimensiona A$ como A$(100,15) el numero en ambas lineas es 100.



PUBLICACION SINTAX

CASILLA DE CORREO Nº 641

1900 - LA PLATA - BUENOS AIRES